

ALLAN CONSELVAN DE OLIVEIRA

MÉTODOS PARA MEDIÇÃO DE FASE E MAGNITUDE DE SINAIS VISANDO
IMPLEMENTAÇÕES EM FPGA

CURITIBA
2015

ALLAN CONSELVAN DE OLIVEIRA

MÉTODOS PARA MEDIÇÃO DE FASE E MAGNITUDE DE SINAIS VISANDO
IMPLEMENTAÇÕES EM FPGA

Dissertação apresentada como requisito parcial
para a obtenção do grau de Mestre, no
Programa de Pós-Graduação em Engenharia
Elétrica da Universidade Federal do Paraná.

Orientador: Prof. PhD. André Mariano

Co-orientador: Prof. Dr. João Dias

CURITIBA

2015

O48m

Oliveira, Allan Conselvan de
Métodos para medição de fase e magnitude de sinais visando
implementações em FPGA/ Allan Conselvan de Oliveira. – Curitiba, 2015.
178 f. : il. color. ; 30 cm.

Dissertação - Universidade Federal do Paraná, Setor de Tecnologia,
Programa de Pós-graduação em Engenharia Elétrica, 2015.

Orientador: André Mariano – Co-orientador: João Dias.
Bibliografia: p. 167-171.

1. Impedância (Eletricidade). 2. Física - Simulação por computador. 3.
Medidas elétricas - Instrumentos. I. Universidade Federal do Paraná.
II. Mariano, André. III. Dias, João . IV. Título.

CDD: 621.37



UNIVERSIDADE FEDERAL DO PARANÁ
Programa de Pós-Graduação em Engenharia Elétrica
PPGEE
Setor de Tecnologia

ATA DE DEFESA DE MESTRADO

Aos dezoito dias do mês de março de 2015, na Sala PK 07 do Departamento de Engenharia Elétrica, foi instalada pela Prof^a. Dr^a. Thelma Solange Piazza Fernandes, Coordenadora do Programa de Pós-Graduação em Engenharia Elétrica, a Banca Examinadora para a centésima sexagésima nona Dissertação de Mestrado do PPGEE, na Área de Concentração em **SISTEMAS ELETRÔNICOS**. Estiveram presentes no ato, além da Coordenadora do Curso de Pós-Graduação, professores, alunos e visitantes.

A Banca Examinadora, atendendo determinação do Colegiado do Programa de Pós-Graduação em Engenharia Elétrica, ficou constituída pelos professores doutores **André Augusto Mariano (Orientador - UFPR)**; **João da Silva Dias (Co-orientadora - UFPR)**; **Sérgio Francisco Pichorim (UTFPR)**; **Luis Henrique Assumpção Lolis (UFPR)** e **Cesar Augusto Dartora (UFPR)**.

Às 09:00 horas, a banca iniciou os trabalhos, convidando o candidato **ALLAN CONSELVAN DE OLIVEIRA** a fazer a apresentação da dissertação intitulada **"MÉTODOS PARA MEDIÇÃO FE FASE E MAGNITUDE DE SINAIS VISANDO IMPLEMENTAÇÃO EM FPGA"**. Encerrada a apresentação, iniciou-se a fase de arguição pelos membros participantes.

Tendo em vista a dissertação e a arguição, a banca decidiu pela:

☒ **APROVAÇÃO**

☐ **NÃO APROVAÇÃO** do candidato, (de acordo com a determinação da Resolução 63/03-CEPE).

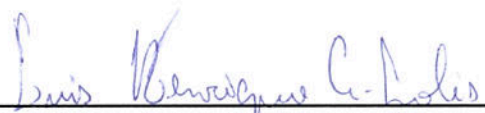
Curitiba, 18 de março de 2015.


Prof. Dr. **André Augusto Mariano**


Prof. Dr. **Sérgio Francisco Pichorim**


Prof. Dr. **João da Silva Dias**


Prof. Dr. **Cesar Augusto Dartora**


Prof. Dr. **Luis Henrique Assumpção Lolis**



AGRADECIMENTOS

Agradeço primeiramente a Deus pela permissão da vida, e a Meishu-Sama pela sua filosofia que permitiu a minha elevação espiritual ao longo do desenvolvimento deste trabalho. Em seguida a meus pais por terem me dado a vida, e a toda a minha família pelo apoio durante os dois anos nos quais realizei este trabalho. À minha noiva Roberta Martini pela paciência em aguentar a grande distância entre Curitiba e São José dos Campos e os poucos encontros mensais, e a meu amigo Paulo Roberto Martini e sua esposa Rosemari Martini, por terem me aguentado lá no Forqueta nos dois últimos verões.

Agradeço também ao meu orientador, o professor PhD André Augusto Mariano, por ter me recebido como orientado como mestrando em Engenharia Elétrica mesmo sendo eu um engenheiro mecânico, e a meu co-orientador, o professor Dr. João dias por ter me auxiliado com as questões relativas a bioimpedância.

Um especial agradecimento ao aluno de iniciação científica Raphael Christian Marins Pereira por ter me ajudado a compreender e dominar o Xlink e o System Generator e a gerar código VHDL, tecnologia sem as quais não seria possível a execução desta pesquisa. Por fim, agradeço também ao aluno de iniciação científica, Cleberton Pereira que me ajudou na pesquisa e teste de circuitos analógicos de fonte de corrente para bioimpedância.

RESUMO

Procurando-se uma solução para determinação de impedância complexa em aplicações de medição de bioimpedância SFBIA (*Single Frequency Bioelectrical Analysis*) em 50 kHz, foram identificadas sete técnicas genéricas para a medição de fase e magnitude de sinais. As técnicas identificadas foram: ajuste de elipse, ajuste de senóide, transformada discreta de Fourier, demodulação de quadratura, modulação de largura de pulso, variação da técnica da figura de Lissajous e medição digital direta por contagem de pulsos. Todas as técnicas foram implementadas em modelo Simulink para verificação da resolução angular possível de ser medida e de possíveis limitações no domínio de utilização. Os resultados iniciais mostraram que apenas a técnica de modulação de largura de pulso não apresentou capacidade de representar a resolução angular estabelecida de $0,1^\circ$ e que a técnica do ajuste de elipse não consegue prever de forma correta fases próximas de 0° e de 180° . As sete técnicas foram parametrizadas de acordo com o uso de recursos lógicos tais como *latches*, memória RAM, memória ROM, frequências de *clock* envolvidas e número total de operações matemáticas necessárias. As operações matemáticas consideradas foram: adição, subtração, multiplicação, divisão, radiciação, seno e arco-seno. A partir dos dados de utilização de pulsos de *clock* por tipo de operação matemática, de um *ip core* de aritmética de ponto flutuante FPU100 versão 19, foi determinada a demanda computacional em número de pulsos de *clock*, para cada uma das técnicas identificadas, considerando-se a utilização de um único *ip core* e sem o uso de *pipelining*. Verificou-se através da comparação de utilização de recursos e da demanda computacional que a demodulação de quadratura é o método com melhor custo benefício dentre os métodos com imunidade a ruído, e que o método da contagem de pulsos é o método com melhor custo benefício dentre os métodos sem imunidade a ruído. Esses dois métodos foram implementados em VHDL e simulados através da *toolbox* System Generator para Simulink, em um modelo comportamental Simulink, que integra o código VHDL e os blocos nativos do Simulink. Os resultados da simulação via emulação em *software* realizaram a verificação do código VHDL frente ao modelo teórico implementado inicialmente. A co-simulação de *hardware*, na qual a simulação do código VHDL ocorre pela execução do código VHDL sintetizado dentro do próprio FPGA, permitiu a validação da técnica da contagem de pulsos. Porém, para a técnica da demodulação de quadratura, verificou-se que o processo de gerenciamento da troca de dados entre o Simulink e o FPGA pelo System Generator, introduziu erros na utilização do passo de tempo máximo do *so/ver* do Simulink, permitindo a validação da técnica apenas no intervalo de defasagem $[10^\circ, 160^\circ]$.

Palavras-chave: bioimpedância, medição de fase, SFBIA, DFT, VHDL, co-simulação de hardware, Simulink, system-generator, FPGA, contagem de pulsos, demodulação de quadratura, Radix-4, ajuste de elipse, ajuste de senóide, figura de Lissajous, PWM, modulação de largura de pulso;

ABSTRACT

In the research for a complex impedance determination solution for SFBIA (Single Frequency Bioelectrical Analysis) 50 kHz bioimpedance measurement applications, seven phase and magnitude signal usual measurement techniques had been identified. These techniques are: ellipse fit, sinusoidal fit, discrete Fourier transform, quadrature demodulation, pulse width modulation, direct digital measurement by pulse counting and a Lissajous figure variety. All these seven techniques had been implemented via Simulink mathematical models for phase angular resolution verification and mathematical limitations identification. The initial results showed that only the pulse width modulation technique was not able to represent the established 0.1° angular resolution, also, the ellipse fit technique could not predict correctly phase values around 0° and 180° . The seven techniques were parameterized according their use of logical resources such as latches, RAM, ROM, clock frequencies and the total number of required mathematical operations. The mathematical operations that were considered are: addition, subtraction, multiplication, division, root extraction, sine and arcsine. Based on the data of clock pulses usage by type of mathematical operation from an ip core of floating-point arithmetic, FPU100 version 19, the computational demand in number of clock cycles for each of the previously identified techniques were estimated. It was considered the use of a single FPU100 ip core and no pipelining. The comparison based on the logical resources utilization and the computational demands have shown that regarding the hardware cost, the quadrature demodulation method is the best one among all the others noise immunity methods tested, and that, the method of pulse counting is the best one regarding hardware usage between the methods with no noise immunity tested. These two methods were implemented via VHDL and simulated through the system-generator toolbox for Simulink, using a behavioral model which integrated the VHDL code with the Simulink native blocks. The results obtained via the software emulation of VHDL code verified the same results obtained with the initial Simulink mathematical models. The hardware cosimulation, in which the synthesized VHDL code is executed inside the FPGA itself, validated the pulse counting method. However, it was found that during the quadrature demodulation cosimulation, in the data exchange management process between the Simulink and the FPGA, the system-generator introduced errors in the maximum step size value of the Simulink solver allowing the validation of phase values only in the phase range of $[10^\circ, 160^\circ]$.

Keywords: bioimpedance, phase difference measurement, FPGA, DDM, discrete Fourier transform, quadrature demodulation, pulse width modulation, Lissajous figure, SFBIA, Simulink, FPGA, system-generator, hardware cosimulation;

LISTA DE FIGURAS

Figura 1 - Composição estrutural da membrana celular.....	20
Figura 2 - a) Impedâncias de uma célula; b) Circuito RC equivalente; c) Circuito equivalente com impedâncias complexas.	21
Figura 3 - a) Passagem da corrente através das células de acordo com a frequência; b) Exemplo de resposta em frequência das correntes normalizadas do modelo RC da célula.	23
Figura 4 - Sistema genérico de medição de bioimpedância	27
Figura 5 - a) Figura de Lissajous resultante da composição de dois sinais; b) Detalhe das linhas de tangência para determinação da relação de frequências.	31
Figura 6 - Figura de Lissajous para dois sinais senoidais de mesma frequência	33
Figura 7 - Comparação das elipses de um sinal com offset e outro sem offset.....	35
Figura 8 - Algoritmo MATLAB de ajuste de elipse proposto por Halir e Flusser.	41
Figura 9 - Diagrama em blocos de um demodulador de quadratura	45
Figura 10 - Relação de fase entre dois sinais	49
Figura 11 - Diagrama em blocos de um medidor de fase.....	51
Figura 12 - Saída do flip-flop RS para um sinal V_{Pt} atrasado em 30°	52
Figura 13 - Saída do flip-flop RS para um sinal V_{Pt} adiantado em 30°	53
Figura 14 - Sinais de entrada e saída de um detector de fase de porta XOR.....	54
Figura 15 - Modelo Simulink de um detector de fase com porta XOR	55
Figura 16 - Resposta de fase para o modelo Simulink do detector de fase com porta XOR.....	55
Figura 17 - Modelo Simulink de um detector de fase com porta XOR e indicação de avanço ou atraso de fase	56
Figura 18 - Forma de onda de um sinal PWM típico com largura de pulso fixa	57
Figura 19 - Distribuição espectral de um sinal PWM.	57
Figura 20 - Comparação de largura de banda entre o filtro passa-baixa e o sinal PWM	58
Figura 21 - Modelo Simulink de um detector de fase completo	60
Figura 22 - Diagrama esquemático de um extrator de fase com correção de temporização.	61
Figura 23 - Formas de onda do extrator de fase, para avanço e atraso de do sinal senoidal defasado	63
Figura 24 - Comparação de topologia de filtros de segunda ordem.....	63
Figura 25 - Diagrama generalizada para impedâncias.	64
Figura 26 - Arranjo experimental no Simulink para simulação e verificação dos métodos de medição de magnitude e fase.	73
Figura 27 - Resposta da função arco-seno para o intervalo $0,2\pi$	74
Figura 28 - Resposta da função arco-tangente para o intervalo $0,2\pi$	75
Figura 29 - Resposta da função arco-cosseno para o intervalo $0,2\pi$	75
Figura 30 - Resposta da função arco-tangente de quatro quadrantes para o intervalo $0,2\pi$	76
Figura 31 - Modelo do detector de magnitude de pico e componente CC.....	77
Figura 32 - Medidor de fase pela técnica de Lissajous.....	80

Figura 33 - Fase estimada pelo modelo Simulink para a técnica derivada da técnica da figura de Lissajous	82
Figura 34 - Erro de estimativa de fase obtida pelo modelo Simulink para a técnica derivada da técnica da figura de Lissajous	82
Figura 35 - Primeiro nível do modelo Simulink de medição de fase por ajuste da elipse	83
Figura 36 - Detalhes internos do bloco do algoritmo de ajuste da elipse	84
Figura 37 - Código MATLAB do bloco "EIG M-file (level-2) S-Function"	85
Figura 38 - Diagrama interno do bloco "Determina Parâmetros"	86
Figura 39 - Fase estimada pelo modelo Simulink para a técnica derivada da técnica do ajuste de elipse.....	87
Figura 40 - Sentido de fase estimado pelo modelo Simulink para a técnica derivada da técnica do ajuste de elipse	88
Figura 41 - Erro da fase estimada pelo modelo Simulink para a técnica derivada da técnica do ajuste de elipse	88
Figura 42 - Componente CC do sinal defasado estimado pelo modelo Simulink para a técnica derivada da técnica do ajuste de elipse	89
Figura 43 - Componente CC do sinal de referência estimado pelo modelo Simulink para a técnica derivada da técnica do ajuste de elipse	89
Figura 44 - Modelo Simulink para medição de fase através de demodulação de quadratura.....	90
Figura 45 - Amplitude estimada pelo demodulador de quadratura para a variação da ordem do filtro passa baixa FIR	92
Figura 46 - Fase estimada pelo demodulador de quadratura para a variação da ordem do filtro passa baixa FIR	92
Figura 47 - Erro percentual dos desvios máximos e mínimos de magnitude e fase para ordem de filtro próxima a 40	93
Figura 48 - Fase estimada pelo modelo Simulink pela técnica de demodulação de quadratura.....	93
Figura 49 - Erro da fase estimada pelo modelo pela técnica da demodulação de quadratura.....	94
Figura 50 - Magnitude estimada pelo modelo para a técnica da demodulação de quadratura.....	94
Figura 51 - Erro percentual para a estimativa de magnitude pela técnica da demodulação de quadratura	95
Figura 52 - Conceito da medição de fase direta digital capaz de medir tanto sinais adiantados quanto sinais atrasados.....	96
Figura 53 - Modelo Simulink para mediação de fase através da medição digital direta.	96
Figura 54 - Fase estimada pelo modelo de medição direta digital	98
Figura 55 - Erro da estimativa de fase do modelo Simulink para a técnica de medição direta digital de fase.....	98
Figura 56 - Modelo Simulink da topologia Lawson-Brown para a medição de diferença de fase através de PWM	100
Figura 57 - Topologia interna do bloco "Chave Analógica" do modelo Simulink da topologia Lawson-Brown.	100
Figura 58 - Tensão média na saída dos filtros	101
Figura 59 - Variação da amplitude da tensão de ripple na saída dos filtros	102

Figura 60 - Retas ajustadas para o filtro de primeira ordem: a) Tensão como função da fase; b) Fase como função da tensão.	103
Figura 61 - Retas ajustadas para o filtro de segunda ordem: a) Tensão como função da fase; b) Fase como função da tensão.	103
Figura 62 - Pontos de discretização da tensão para o intervalo de resolução angular $0,1^\circ$, 1° com passo de $0,1^\circ$	104
Figura 63 - Modelo Simulink para o teste da técnica de medição de razão magnitude e fase através da técnica de DFT	107
Figura 64 - Trecho do código responsável pela funcionalidade dos blocos do tipo "Maior Magnitude - L2MSF"	107
Figura 65 - Comparação da resposta entre o algoritmo Radix-4 e o algoritmo Radix-2 para o ângulo de 50°	109
Figura 66 - Fase estimada pelo modelo FFT para 1024 pontos de amostragem	110
Figura 67 - Razão de amplitudes estimada pelo modelo FFT para 1024 pontos de amostragem.....	110
Figura 68 - Fase estimada pelo modelo FFT para 256 pontos de amostragem	111
Figura 69 - Fase estimada pelo modelo FFT para 64 pontos de amostragem.....	111
Figura 70 - Fase estimada pelo modelo FFT para 16 pontos de amostragem.....	112
Figura 71 - Modelo Simulink para medição de fase e magnitude através da técnica do ajuste da senóide	113
Figura 72 - Trecho de script MATLAB para criar a matriz <i>mA</i>	113
Figura 73 - Fase estimada pelo modelo de ajuste de senóide para 256 pontos de amostragem.....	114
Figura 74 - Amplitude do sinal defasado estimado pelo modelo de ajuste da senóide para 256 pontos de amostragem.....	115
Figura 75 - Interface básica do System Generator dentro do Simulink.....	126
Figura 76 - Visão geral do conceito de co-simulação para validação em hardware do código VHDL com processamento de sinais pelo Simulink	127
Figura 77 - Modelo de co-simulação de hardware para a porta E apresentada na Figura 75.....	128
Figura 78 - Topologia do ADC virtual utilizado para simulação.	129
Figura 79 - Modelo System Generator do demodulador de quadratura.....	133
Figura 80 . Resposta do modelo System Generator para o caso de teste padrão: a) Resposta de amplitude; b) Resposta de fase.....	134
Figura 81 - Inserção do detector de componente CC no modelo do demodulador de quadratura.....	135
Figura 82 - Sinal representativo de magnitude: a) Sinal na saída do filtro FIR; b) Sinal na saída do detector CC.....	136
Figura 83 - Sinal representativo de fase: a) Sinal na saída do filtro FIR; b) Sinal na saída do detector CC.....	137
Figura 84 - Resposta do modelo VHDL do demodulador de quadratura após a inserção dos detectores de componente CC.....	138
Figura 85 - Resposta de fase do modelo do demodulador de quadratura com detector de componente CC para o intervalo 0° , 360° com passo de 1°	139
Figura 86 - Resposta de magnitude do modelo do demodulador de quadratura com detector de componente CC para o intervalo 0° , 360° com passo de 1°	139
Figura 87 - Modelo do demodulador de quadratura para validação via co-simulação em hardware.....	140

Figura 88 - Resposta de fase obtida através da co-simulação em hardware do demodulador de quadratura	141
Figura 89 - Resposta de amplitude obtida através da co-simulação em hardware do demodulador de quadratura	141
Figura 90 - Modelo System Generator da medição de fase pela contagem de pulsos	144
Figura 91 - Resposta de fase do modelo System Generator da técnica de contagem de pulsos.....	145
Figura 92 - Resposta de amplitude do modelo System Generator da técnica de contagem de pulsos.....	145
Figura 93 - Modelo final para a co-simulação em hardware da técnica de medição de fase por contagem de pulsos	146
Figura 94 - Resposta de fase da co-simulação em hardware da técnica de contagem de pulsos.....	147
Figura 95 - Resposta de amplitude da co-simulação de hardware da técnica de contagem de pulsos.....	147
Figura 96 - Arranjo para inserção de ruído branco gaussiano.	167
Figura 97 - Resposta SNR da técnica da figura de Lissajous.....	168
Figura 98 - Resposta SNR da técnica do ajuste de elipse	168
Figura 99 - Resposta SNR da técnica da demodulação de quadratura	169
Figura 100 - Resposta SNR da técnica de medição direta digital	169
Figura 101 - Resposta SNR para a técnica PWM.....	169
Figura 102 - Resposta SNR para a técnica DFT - Radix-2	170
Figura 103 - Resposta SNR para a técnica da DFT - Radix-4.....	170
Figura 104 - Resposta SNR para a técnica do ajuste de senóide	170

LISTA DE TABELAS

Tabela 1 - Especificações do caso de estudo.....	28
Tabela 2 - Exemplo de cálculo de frequência de clock dos pulsos de contagem.....	51
Tabela 3 - Utilização de recursos do algoritmo Radix-4 em um FPGA Spartan-3	68
Tabela 4 - Consumo de recursos do FPGA pelo ip core FPU100 Versão 19	116
Tabela 5 - Parametrização do processo lógico-aritmético das técnicas de medição de magnitude e fase simuladas em Simulink.....	119
Tabela 6 - Quantização das operações aritméticas básicas para operações matemáticas compostas.....	120
Tabela 7 - Pulsos de clock necessários para cada operação aritmética do ip core FPU100 versão 19	121
Tabela 8 - Quantidade de operações aritméticas básicas necessárias para o algoritmo CORDIC com 10 iterações.....	121
Tabela 9 - Total de pulsos de clock utilizados por cada técnica de medição.....	122
Tabela 10 - Coeficientes do filtro FIR de ordem 40.....	132
Tabela 11 - Comparação entre as topologias implementadas em VHDL e as topologias reportadas na literatura.....	148
Tabela 12 - Comparação do consumo de recursos no FPGA.....	149
Tabela 13 - Comparação de resposta SNR	171

LISTA DE SIGLAS

ADC	—	<i>Analogic to Digital Converter</i>
ASIC	—	<i>Application Specific Integrated Circuit</i>
ATP	—	<i>Adenosine Triphosphate</i>
BIS	—	<i>Bioelectrical Impedance Spectroscopy</i>
CA	—	Corrente alternada
CC	—	Corrente contínua
CMOS	—	<i>Complementary Metallic Oxide Semiconductor</i>
CORDIC	—	<i>Coordinate Rotation Digital Computer</i>
DDM	—	<i>Direct Digital Measurement</i>
DFT	—	<i>Discrete Fourier Transform</i>
DSP	—	<i>Digital Signal Processor</i>
DQ	—	Demodulador de Quadratura
ECW	—	<i>Extracelular Water</i>
EIT	—	<i>Electrical Impedance Tomography</i>
FFM	—	<i>Free Fat Mass</i>
FFT	—	<i>Fast Fourier Transform</i>
FIR	—	<i>Finite Impulse Response</i>
FPGA	—	<i>Field Programmable Gate Arrays</i>
FPU	—	<i>Float Point Aritmetic Unit</i>
ICW	—	<i>Intracelular Water</i>
IIR	—	<i>Infinite Impluse Response</i>
MBSE	—	<i>Model-based System Engineering</i>
MFBIA	—	<i>Multiple Frequency Bioelectrical Analysis</i>
NAN	—	<i>Not a Number</i>
PSD	—	<i>Phase Sensitive Detector</i>
PWM	—	<i>Pulse Width Modulation</i>
RAM	—	<i>Random-access Memory</i>
ROM	—	<i>Read Only Memory</i>
SFBIA	—	<i>Single Frequency Bioelectrical Analysis</i>
SNR	—	<i>Signal Noise Ratio</i>

USB	—	<i>Universal Serial Bus</i>
TBW	—	<i>Total Body Water</i>
VCCS	—	<i>Voltage Controlled Current Source</i>
VHDL	—	<i>VHSIC Hardware Description Language</i>
VHDLAMS	—	<i>VHSIC Hardware Description Language and Mixed-Signal Extensions</i>
VHSIC	—	<i>Very High Speed Integrated Circuit</i>

SUMÁRIO

1. INTRODUÇÃO	15
1.1 BIOIMPEDÂNCIA E TÉCNICA DE BIOIMPEDÂNCIA	17
1.1.1 MODELO DE IMPEDÂNCIA CELULAR	19
1.1.2 TÉCNICAS DE BIOIMPEDÂNCIA	24
1.1.3 VALORES CARACTERÍSTICOS DE BIOIMPEDÂNCIA	25
1.1.4 SISTEMA GENÉRICO DE MEDIÇÃO DE BIOIMPEDÂNCIA	26
1.1.5 ESPECIFICAÇÕES PARA O CASO DE ESTUDO	28
1.2 OBJETIVOS	29
1.3 OBJETIVOS ESPECÍFICOS	29
1.4 ORGANIZAÇÃO DO TRABALHO	30
2. TÉCNICAS DE MEDIÇÃO DE DIFERENÇA DE FASE E MAGNITUDE	31
2.1 TÉCNICA DA FIGURA DE LISSAJOUS	31
2.2 TÉCNICA DO AJUSTE DA ELIPSE	37
2.3 TÉCNICA DO DEMODULADOR DE QUADRATURA	44
2.4 TÉCNICA DA MEDIÇÃO DIRETA DIGITAL	49
2.5 TÉCNICA DA MEDIÇÃO DE FASE POR PWM	53
2.6 TÉCNICA DA MEDIÇÃO POR DFT	67
2.7 TÉCNICA DA MEDIÇÃO DE FASE POR AJUSTE DE SENOIDE	71
3. SIMULAÇÃO DOS MÉTODOS DE MEDIÇÃO EM SIMULINK	73
3.1 ARRANJO EXPERIMENTAL NO SIMULINK	73
3.2 CONSIDERAÇÕES GERAIS PARA USO DO SIMULINK	74
3.3 DETECÇÃO DE MAGNITUDE DE PICO E DE VALE	77
3.4 SIMULAÇÃO DA TÉCNICA DA FIGURA DE LISSAJOUS	79
3.5 SIMULAÇÃO DA TÉCNICA DO AJUSTE DA ELIPSE	83
3.6 SIMULAÇÃO DA TÉCNICA DE DEMODULAÇÃO DE QUADRATURA COM FILTRO FIR DIGITAL	90
3.7 SIMULAÇÃO DA TÉCNICA DE MEDIÇÃO DIRETA DIGITAL	95
3.8 SIMULAÇÃO DA TÉCNICA DE PWM	99
3.9 SIMULAÇÃO DA TÉCNICA DE DFT	106
3.10 SIMULAÇÃO DA TÉCNICA DO AJUSTE DA SENOIDE	112
3.11 SELEÇÃO DAS TÉCNICAS DE MEDIÇÃO DE FASE PARA IMPLEMENTAÇÃO EM VHDL	115
3.11.1 CRITÉRIOS DE SELEÇÃO	115

3.11.2	REQUISITOS A SEREM CUMPRIDOS PELAS TOPOLOGIA SELECIONADAS	117
3.11.3	COMPARAÇÃO ENTRE AS TÉCNICAS	118
4.	VERIFICAÇÃO DO CÓDIGO VHDL E VALIDAÇÃO EM <i>HARDWARE</i>	125
4.1	SYSTEM GENERATOR	125
4.2	VALIDAÇÃO EM <i>HARDWARE</i> VIA CO-SIMULAÇÃO	127
4.3	MODELAGEM COMPORTAMENTAL DO ADC.....	129
4.4	DEMODULADOR DE QUADRATURA.....	129
4.4.1	DIMENSIONAMENTO DO SCALING	131
4.4.2	VERIFICAÇÃO DO CÓDIGO VHDL.....	133
4.4.3	VALIDAÇÃO EM <i>HARDWARE</i>	140
4.5	MEDIÇÃO DIRETA DIGITAL	142
4.5.1	VERIFICAÇÃO DO CÓDIGO VHDL.....	143
4.5.2	VALIDAÇÃO EM <i>HARDWARE</i>	146
4.6	COMPARAÇÃO DAS IMPLEMENTAÇÕES EM VHDL E A LITERATURA	148
4.7	COMPARAÇÃO DO CONSUMO DE RECURSOS DO FPGA ENTRE O DQ E O DDM.....	149
5.	CONCLUSÕES E TRABALHOS FUTUROS	150
5.1	CONCLUSÕES.....	150
5.1.1	A RESPEITO DOS MÉTODOS DE NUENS DE PONTOS.....	150
5.1.2	A RESPEITO DOS MÉTODOS DE MEDIÇÃO DIRETA	151
5.1.3	A RESPEITO DA VERIFICAÇÃO DO CÓDIGO VHDL	153
5.1.4	A RESPEITO DA VALIDAÇÃO EM <i>HARDWARE</i>	154
5.1.5	A RESPEITO DO CONSUMO DE RECURSOS DO FPGA	154
5.2	TRABALHOS FUTUROS	155
5.2.1	RELACIONADOS AO PROCEDIMENTO DE VERIFICAÇÃO E VALIDAÇÃO DO CÓDIGO VHDL	155
5.2.2	RELACIONADOS AO DEMODULADOR DE QUADRATURA	156
5.2.3	RELACIONADOS AO MÉTODO DDM	157
5.2.4	RELACIONADOS AOS MÉTODOS TEÓRICOS PESQUISADOS.....	158
5.2.5	RELACIONADOS À BIOIMPEDÂNCIA.....	160
5.2.6	RELACIONADOS À ARITMÉTICA DE PONTO FLUTUANTE.....	160
	REFERÊNCIAS	161
	ANEXO A – CÓDIGO MATLAB DO ALGORITMO RADIX-4.....	166
	APÊNDICE A – ANÁLISE SNR DOS MÉTODOS TEÓRICOS	167

1. INTRODUÇÃO

A aplicação mais conhecida para a medição de fase e magnitude de sinais é a medição de impedância. Tradicionalmente sabe-se que um dos usos mais populares da medição de impedância é a caracterização de materiais dielétricos. Como exemplo, o trabalho de Phatungthane et al. (2014) mostra o uso da medição de impedância para caracterizar melhorias de dielétrico para um novo processo de fabricação de cerâmica denominada BFT ($Ba(Fe_{1/2}Ta_{1/2})O_3$). Outra grande aplicação da medição de impedância é o monitoramento de processos eletroquímicos. No trabalho de Lulu et al. (2015) a espectroscopia por medição de impedância é utilizada para a análise da eletrólise despolarizada de SO_2 no ciclo sulfúrico híbrido durante o processo de separação de água para a obtenção de hidrogênio.

Já na indústria petroquímica, conforme mostra Pirouzpanah et al. (2014), a espectroscopia de impedância pode ser utilizada para se determinar a quantidade de água existente em misturas de água e óleo cru, permitindo a determinação do tipo de dispersão da mistura. Ainda, na engenharia civil e na mecânica estrutural, a medição da impedância gerada por transdutores piezoelétricos pode ser utilizada para o monitoramento da saúde estrutural de componentes e vigas conforme mostra Sun et al. (2015).

A medição de magnitude e fase não se restringe apenas à medição da impedância. De acordo com o trabalho de Oliveira (2005), a medição da fase e da amplitude de um sinal de ultrassom pode ser utilizada para a implementação da técnica do “tempo de voo” (o “tempo de voo” é definido como o tempo decorrido entre a transmissão e a recepção de um sinal) para a medição do nível de reservatórios industriais.

Além das aplicações, a determinação da magnitude e fase de sinais encontra aplicações bastante importantes na área biomédica. Um exemplo bastante recente do uso desse princípio é o sensor descartável para

monitoramento de inflamações do sistema nervoso mostrado por Biela et al. (2015). Este sensor possui um filme de hidro gel que reage com a enzima metaloproteinase-9 da amostrada sendo testada. A reação química resultante degrada o filme de hidro gel porém produzindo de forma muito rápida uma variação impedância bastante significativa permitindo a detecção da enzima metaloproteinase-9.

Outro exemplo relevante de aplicação biomédica da medição de impedância é o monitoramento de recém-nascidos através da tomografia por impedância elétrica (EIT). Tendo em vista que na medição de impedância existe a total ausência de emissões radiológicas e de indução de reações químicas no tecido examinado, do ponto de vista de segurança, a EIT é a única técnica de geração de imagens que não agride o sistema imunológico de recém-nascidos. (TRANTIS et al., 2011)

De modo geral, verifica-se na atualidade uma gama bastante vasta de equipamentos eletrônicos na categoria de equipamentos biomédicos que são baseados na técnica da medição de impedância. Esse tipo de equipamento é utilizado tanto em aplicações clínicas quanto em aplicações laboratoriais, principalmente para: a avaliação da composição do corpo, a avaliação da condição nutricional de pacientes, a caracterização de materiais biológicos e a identificação de células cancerígenas. (AL-RAWI et al, 2013)

Os equipamentos baseados na medição de impedância são reconhecidamente vantajosos pois possuem as vantagens diretas da técnica da medição de impedância: é segura, tem baixo custo, a implementação é simples, a resposta é rápida, o processo é não invasivo, e é fácil de aplicar. (AMARAL et al., 2009)

Dados os avanços nas técnicas da medição de impedância em tecidos biológicos e nas soluções de microeletrônica, observa-se atualmente a extensão da utilização dos equipamentos biomédicos para outros setores além do setor das aplicações médicas clínicas e laboratoriais. Dentre os exemplos das aplicações não médicas encontram-se: balanças domésticas que medem o índice de massa gorda e de água do corpo, sistemas de avaliação da qualidade de alimentos perecíveis, como o leite, pesquisa botânica, e sistemas de

monitoramento de cultura microbiológica. (BERTEMES-FILHO et al, 2010; BLACK&DECKER, 2012; LIN et al., 2012; WISSENWASSER et al., 2011)

Como tecnologia, a medição de impedância em tecidos biológicos passou a ser denominada como técnica de bioimpedância, e atualmente, ela é um dos principais métodos para a obtenção de dados biológicos através de sistemas eletrônicos. No cenário de aplicação da tecnologia, observa-se vasta utilização da correlação de parâmetros biológicos com valores de bioimpedância em diversos segmentos de aplicação, tais como médico, laboratorial, controle de produto e eletrodomésticos.

Segundo Isaacson (1986), nos equipamentos destinados a EIT os valores de tensão e corrente mínimos distinguíveis, são o ponto crítico para a obtenção da confiabilidade de diagnóstico. O desenvolvimento de circuitos integrados analógicos para instrumentação e circuitos integrados para o processamento digital de sinais (DSP), permitiu a implementação de técnicas de supressão de ruídos e fontes de corrente com controle mais preciso de frequência e intensidade. A utilização dos avanços da microeletrônica nos equipamentos de bioimpedância, permitiu a fabricação de equipamentos mais precisos, especialmente os destinados a EIT. No cenário atual, observa-se a tendência de integração de soluções de medição de bioimpedância em circuitos integrados de aplicação específica (ASIC), como é apresentado no trabalho de Hussain (2012).

Dentre todas as funcionalidades que fazem parte da integração dentro de uma solução para a medição de bioimpedância, a determinação da fase e da magnitude da impedância é a essência da solução. Utilizando-se das ferramentas de *hardware* reconfigurável se é possível desenvolver uma solução de medição de componentes complexas (magnitude e fase) com agilidade e confiabilidade, seguindo a tendência atual de integração da solução de medição de bioimpedância em circuitos integrados de aplicação específica.

1.1 BIOIMPEDÂNCIA E TÉCNICA DE BIOIMPEDÂNCIA

A técnica da bioimpedância é definida como a determinação de parâmetros fisiológicos com base nas propriedades elétricas de tecidos

biológicos (vivos, em processo de morte ou já completamente mortos) aplicando-se uma corrente elétrica, medindo-se a resposta a essa corrente elétrica aplicada, e realizando-se a inferência dos parâmetros biológicos relacionados a essa resposta. (BELMONT et al., 2013)

O princípio da técnica baseia-se no fato de que as células têm na sua composição água e sais. Os sais presentes na célula se dissociam em íons, e esses íons, conseqüentemente, formam eletrólitos, os quais permitem em certo grau a passagem da corrente elétrica através da célula. A oposição à passagem da corrente elétrica no tecido biológico, que é regulada pela concentração dos íons nas células, denomina-se bioimpedância.

O tecido biológico é composto por uma imensa quantidade de células, as quais estão imersas dentro do mesmo fluido extracelular. A membrana celular tem condutividade elétrica praticamente nula devido a sua composição química. A sua pequena condutividade elétrica é resultado de estruturas transpassantes relacionadas a permeabilidade seletiva, a qual controla o fluxo de íons entre o fluido intracelular e o fluido extracelular.

O fluido intracelular tem potencial elétrico não nulo e está isolado do fluido extracelular pela membrana celular (a permeabilidade seletiva da membrana celular pode isolar os meios ou permitir o fluxo de íons). O fluido extracelular também possui potencial elétrico não nulo. Assim, existe uma diferença de potencial elétrico entre o fluido intracelular e o fluido extracelular. Aproximando-se o de tecido biológico como uniforme, todas as células apresentam o mesmo potencial elétrico no fluido intracelular assim como compartilham a mesma interface com o fluido extracelular.

Como resultado da hipótese de se considerar o tecido uniforme, todas as células apresentam a mesma diferença de potencial elétrico constituindo numa associação em paralelo. Conseqüentemente, o mesmo modelo elétrico que descreve a condutividade de uma única célula pode ser utilizado para descrever a condutividade elétrica geral de todo o tecido, sendo que os parâmetros de impedância são equivalentes ao resultado da associação das impedâncias de cada célula que compõem o tecido.

Na prática o tecido biológico não é uniforme, assim, para casos de caracterização de parâmetros intracelulares, utiliza-se a teoria de mistura de Hanai, a qual descreve o efeito de condutividade de uma suspensão de esferas não-condutoras que estão imersas dentro de um meio condutor (uma vez que as células são eletricamente isoladas pela membrana celular, elas são aproximadas como esferas não condutoras). (STAHN et al., 2012)

O tecido biológico pode ser passivo ou ativo. A grande maioria das aplicações da técnica de bioimpedância é feita sobre tecidos predominantemente passivos nos quais a membrana celular se comporta como um capacitor em carga e descarga sob a ação do estímulo elétrico. Porém, quando um estímulo elétrico é aplicado a um tecido ativo, o estímulo provoca a geração de um potencial de ação dentro da célula. O potencial de ação faz com que a célula se polarize até um pico cuja intensidade elétrica é muito superior a intensidade do estímulo aplicado. Como resultado, o estímulo externo para de surtir efeito. Após isso, a ATP (*Adenosine Triphosphate*) produz uma dissociação de íons causando a repolarização da célula até o potencial de repouso, restaurando a sensibilidade ao estímulo elétrico externo.

O modelo elétrico que descreve o comportamento de um tecido ativo é um modelo de condutância paralela, cujos valores de condutância dependem da variação temporal da probabilidade de abertura e fechamento dos canais permeáveis da membrana celular. Essa probabilidade reflete então, a probabilidade de cada canal ativo da membrana conseguir abrir e fechar controlando a movimentação de íons entre o fluido intracelular e o fluido extracelular. Esse modelo de condutância paralela para tecidos ativos é conhecido como modelo de Hodgkin-Huxley. Detalhes sobre a modelagem de uma membrana ativa podem ser vistos em Plonsey e Barr (2007).

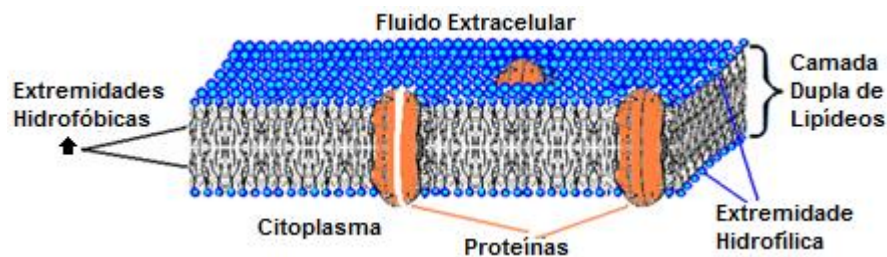
1.1.1 MODELO DE IMPEDÂNCIA CELULAR

Um tecido biológico qualquer é constituído de inúmeras células imersas dentro de um fluido extracelular. Cada célula é delimitada por uma membrana, possuindo em seu interior um fluido intracelular denominado citoplasma. O fluido

extracelular e o fluido intracelular são compostos de eletrólitos e proteínas. Segundo Martinez (2007), a membrana celular se constitui de uma estrutura fina e elástica cuja espessura varia de 75 até 100 angstroms, e é composta primariamente por aproximadamente 55% de proteínas e 43% de lipídeos. Os lipídeos conferem a forma estrutural à membrana e estão dispostos em uma pilha de duas camadas (camada dupla de lipídeos) conforme mostrado na

Figura 1.

Figura 1 - Composição estrutural da membrana celular



Fonte: (MARTINEZ, 2007)

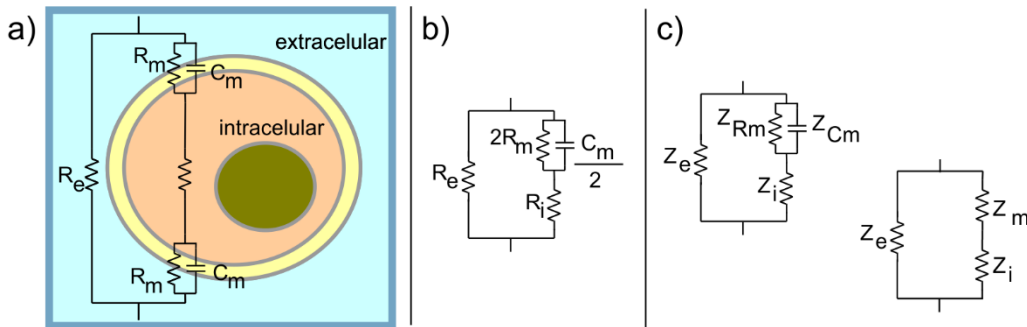
As moléculas de lipídeos que compõem a membrana celular possuem uma extremidade hidrofílica (esferas azuis na Figura 1), a qual fica voltada ou para o meio intracelular ou para meio extracelular, e uma extremidade hidrofóbica a qual fica voltada para o interior da membrana. As proteínas que compõem a membrana celular estão espalhadas dentro da estrutura de lipídeo e formam os canais passivos e canais ativos. Os canais passivos estão sempre abertos e os íons os atravessam de acordo com os gradientes químicos e os gradientes elétricos entre o fluido intracelular e o fluido extracelular. Os canais ativos dependem de estímulos químicos e elétricos para que permitam a passagem dos íons.

Os lipídeos que compõem a estrutura da membrana celular são isolantes, porém, tanto o fluido intracelular quanto o fluido extracelular possuem carga elétrica. Deste modo, o lipídeo da membrana celular se comporta como um material dielétrico entre os dois fluidos, fazendo com que a interface (fluido extracelular, membrana celular, fluido intracelular) crie um capacitor orgânico.

Os canais passivos presentes na membrana celular permitem uma baixa movimentação de íons entre o meio intracelular e o meio extracelular, resultando em uma alta resistência elétrica associada em paralelo ao capacitor orgânico. Tanto o fluido intracelular quanto o fluido extracelular são eletrólitos. Como os eletrólitos apresentam oposição a passagem da corrente elétrica, os fluidos intracelular e extracelular formam resistências elétricas.

Para um tecido passivo, o comportamento elétrico celular pode ser modelado através de um circuito RC conforme mostra a Figura 2.

Figura 2 - a) Impedâncias de uma célula; b) Circuito RC equivalente; c) Circuito equivalente com impedâncias complexas.



Adaptado de: (CERÁVOLO, 2012)

Considerando-se as impedâncias relativas aos resistores e ao capacitor do circuito RC que modela a célula, a impedância equivalente é expressa pela equação (1), na qual, Z_e é a impedância do fluido extracelular, Z_i é a impedância do fluido intracelular, Z_{Rm} é impedância resistiva da membrana celular e Z_{Cm} a impedância capacitiva da membrana celular. Denotando-se a associação paralela entre Z_{Cm} e Z_{Rm} por Z_m , equação (2), a impedância equivalente Z_a é obtida pela equação (3).

$$Z_{equivalente} = Z_e \parallel (Z_i + Z_{Rm} \parallel Z_{Cm}) \quad (1)$$

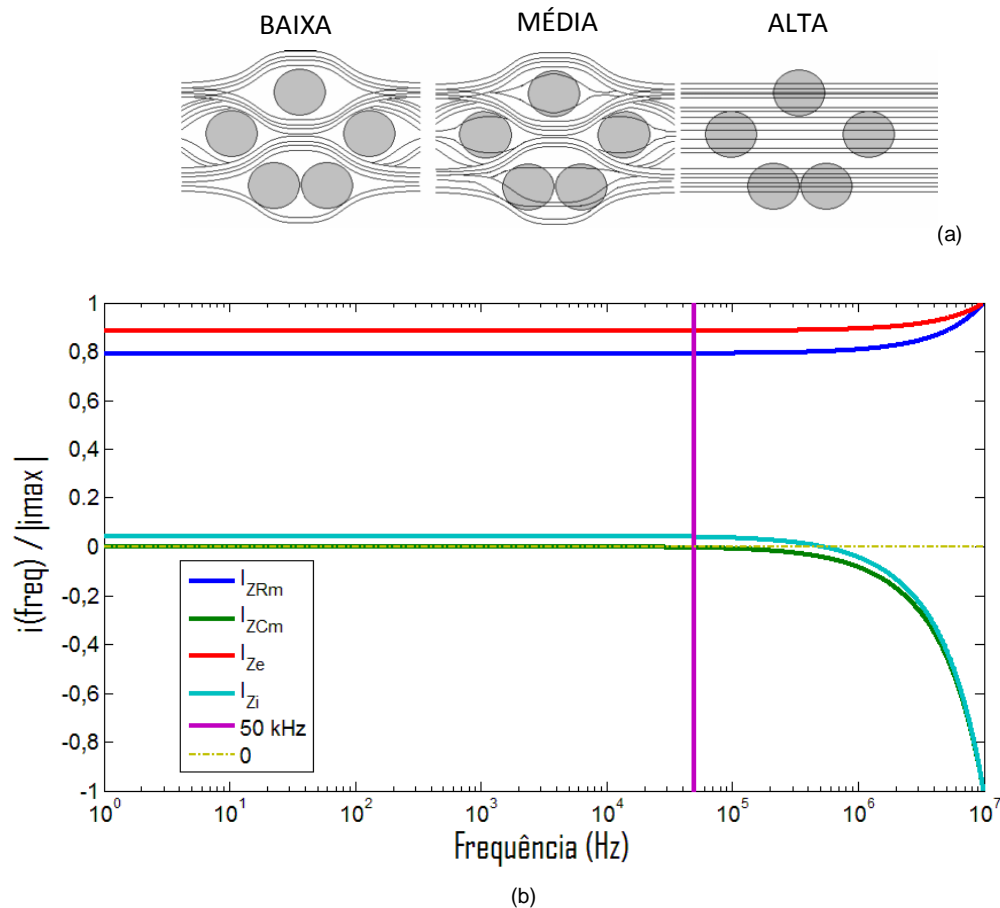
$$Z_m = Z_{R_m} \parallel Z_{C_m} = \frac{Z_{R_m} \cdot Z_{C_m}}{Z_{R_m} + Z_{C_m}} \quad (2)$$

$$Z_a = Z_e \parallel (Z_i + Z_m) \quad (3)$$

Dado o comportamento capacitivo da membrana celular, se a corrente aplicada à célula é uma corrente CC ou uma corrente CA de baixa frequência, o efeito capacitivo da membrana impede que a corrente elétrica a atravesse. Nesse caso a condução elétrica ocorre pelo fluido extracelular. À medida que a frequência aumenta, menor será a reatância capacitiva da membrana celular. Como consequência, a corrente que era conduzida apenas pelo meio extracelular, passa a atravessar a membrana celular proporcionalmente ao aumento da frequência. Para altas frequências a membrana celular age como uma baixa impedância, e assim, o efeito capacitivo da membrana desaparece e a corrente flui tanto ao redor das células (através do fluido extracelular), quanto pelo interior das células. Porém, ainda regulada pela condutividade iônica do fluido intracelular. (GRIMNES, MARTINSEN, 2008)

A Figura 3a ilustra o fluxo de corrente através das células para as situações de baixa, média e alta frequência. A Figura 3b mostra a resposta em frequência das correntes normalizadas que passam através das células. Essas correntes são resultado de um modelo RC cujos valores de resistência e capacitância foram obtidos do caso padrão de teste da plataforma de medição de gordura corporal por bioimpedância proposta por Cerávolo (2012). Os valores de resistência e capacitância são: $R_e = 560 \, \Omega$, $R_i = 560 \, \Omega$, $R_m = 100 \, k\Omega$, $C_m = 3 \, pF$. Esses valores correspondem aos elementos resistivos e capacitivos da Figura 2b e devem ser utilizados para se calcular as impedâncias mostradas na Figura 2c.

Figura 3 - a) Passagem da corrente através das células de acordo com a frequência; b) Exemplo de resposta em frequência das correntes normalizadas do modelo RC da célula.



Fonte da figura 3a: (MARTINEZ, 2007)

A Figura 3b mostra as correntes normalizadas para o espectro de 0Hz (corrente CC) até 10 MHz, obtidas com a aplicação de uma corrente de 1 mA no circuito da Figura 2b. Cada corrente normalizada é calculada pela razão entre seu valor na dada frequência, e o maior módulo da corrente encontrado em todo o espectro de frequência. Observa-se que as correntes normalizadas que fluem através do interior da célula, I_{Z_i} , e da impedância capacitiva da membrana, $I_{Z_{Cm}}$, mantêm-se próximas de zero até uma frequência próxima de 1 MHz. Após esse valor de frequência, o valor da corrente através da membrana e no interior da célula crescem significativamente, conforme o efeito mostrado na Figura 3a para

altas frequências. Observa-se também, que a corrente através do fluido extracelular, I_{Ze} , e a corrente na superfície da membrana celular, I_{ZRm} , são próximas do valor máximo durante todo o espectro. Isso indica a condução da corrente elétrica através do fluido extracelular durante todo o espectro.

1.1.2 TÉCNICAS DE BIOIMPEDÂNCIA

Atualmente existem quatro técnicas de bioimpedância, cujas diferenças entre si consistem dos parâmetros fisiológicos que são correlacionados com a bioimpedância medida, a quantidade de frequências utilizadas no processo de medição, e, a quantidade de segmentos localizados no corpo sob exame, onde são tomadas as medições. (KYLE et al, 2004)

A primeira técnica que foi desenvolvida e é amplamente utilizada é a técnica SFBIA (*single frequency bioelectrical analysis*). Ela se baseia na aplicação de uma corrente alternada de 50 kHz ao material biológico e é bastante utilizada para levantar a condição nutricional de pacientes, pois permite medir a quantidade total de água no corpo uma vez que a corrente passa tanto pelo fluido intracelular quanto pelo fluido extracelular. A partir de uma correlação empírica é possível se determinar a massa magra a partir da quantidade total de água do corpo. Como desvantagem, essa técnica não permite o estudo de diferentes fluidos intracelulares pois não consegue medir a quantidade de água intracelular isoladamente.

Dada à experimentação da resposta de tecidos biológicos à uma gama maior de frequências, criou-se a técnica da MFBIA (*multiple frequency bioelectrical analysis*). Nessa técnica são executadas medição com cinco a sete frequências, sendo normalmente utilizados os valores: CC, 1 kHz, 5 kHz, 50 kHz, 100 kHz, 200 kHz a 500 kHz. A MFBIA permite a avaliação da massa magra, a quantidade total de água do corpo, a quantidade de água extracelular e quantidade de água intracelular. Para frequências abaixo de 5 kHz e para frequências superiores a 200 kHz, os resultados não são representativos. (KYLE et al, 2004)

Na busca por se tentar determinar com maior precisão a caracterização celular, criou-se a BIS (*bioelectrical impedance spectroscopy*). Nessa técnica são feitas a medição em 256 frequências diferentes (valores entre 5 kHz e 1 MHz) e através de modelos matemáticos são calculadas as impedâncias resistivas na frequência 0 Hz e numa frequência muito alta, denominada frequência infinita. Conforme mostra Lorenzo et al. (1997), esses dois valores são utilizados no diagrama de Cole-Cole e na teoria mista de Hanai para se determinar a massa magra e a massa gorda com precisão. A impedância na frequência zero é bastante representativa da medida da quantidade de água extracelular, enquanto que a resistência na frequência infinita é confiável para a determinação da quantidade de água total do corpo. (IMPEDMED, 2013)

Segundo Triantis et al. (2011), dada a necessidade de geração de imagem para diagnósticos com métodos que não utilizassem a emissão de energia foi desenvolvida a técnica da EIT (*electrical impedance tomography*), a qual tem aplicação principalmente no acompanhamento de recém nascidos. De acordo com Holder (2004), a EIT é promissora nas aplicações de monitoramento de função gastrointestinal, função cardíaca e imagem torácica dos pulmões, detecção de câncer de mama e imagem de função cerebral.

Os tomógrafos de bioimpedância utilizam uma matriz de eletrodos a qual é aplicada na superfície de um volume, permitindo que um campo eletromagnético interno ao corpo se estabeleça, produzindo tensões e correntes, as quais são medidas e utilizadas para se criar uma imagem relacionada a espessura do corpo. Os equipamentos de tomografia por bioimpedâncias atuais utilizam frequências de até 10 MHz com diversos tipos de formas de onda. (PORTO, 2009)

1.1.3 VALORES CARACTERÍSTICOS DE BIOIMPEDÂNCIA

Segundo a revisão de referências de valores apresentada por Porto (2009), a maioria dos tecidos humanos com exceção de osso e gorduras, apresenta uma resistividade variando de $150 \, \Omega \cdot \text{cm}$ a $675 \, \Omega \cdot \text{cm}$. Ainda de acordo com Porto (2009), a consulta a diversos estudos de determinação do

índice de gordura do corpo através das técnicas de bioimpedância, indicam que o corpo de uma pessoa saudável apresenta em média, uma impedância total entre 200 Ω a 800 Ω , sendo que o valor mais frequente é de 500 Ω .

Segundo Holder (2004) apud Porto (2009), para aplicações de tomografia de bioimpedância, os valores das impedâncias elétricas relacionadas as regiões medidas tem resistividade tipicamente entre 100 Ω e 10 k Ω .

De acordo com o manual do equipamento padrão de medição de SFBIA da BIODYNAMICS, *Bioimpedance Analyzer model 450*, a medição de reatância capacitiva do equipamento é na ordem de 0 Ω a 300 Ω e a medição de resistividade da ordem e 0 Ω a 1500 Ω . A frequência utilizada é de 50 kHz e a faixa de medição angular é de 0° a 20° com resolução de 0,1°. (BIODYNAMICS-CORPORATION, 2014)

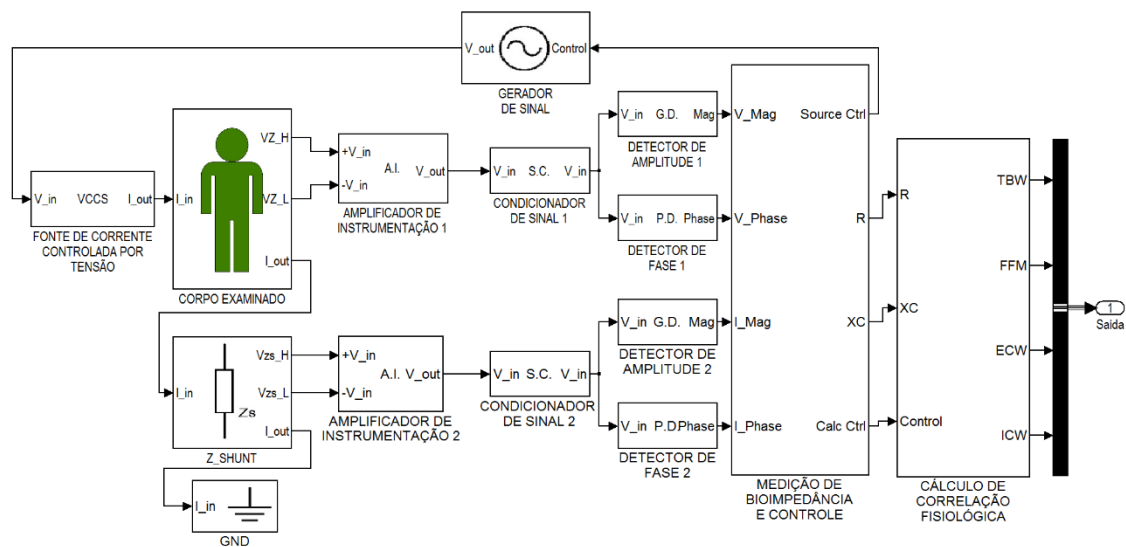
Segundo comparação apresentada por Grimnes e Martinsen (2008), os valores de reatância em coordenada polar podem chegar a 30° para músculos e para a pele humana molhada, sendo os valores mais comuns chegando a 20°.

O valor da corrente aplicada ao tecido varia de acordo com a aplicação. Para implantes biomédicos as correntes são da ordem de 10 μ A enquanto que para as aplicações de medição de quantidade total de água pode chegar à 800 μ A. A corrente utilizada deve via de regra ser inferior à 1 mA, pois esse valor, é o limiar da sensação de choque elétrico para corrente alternada. (FRITSCH et al, 2008)

1.1.4 SISTEMA GENÉRICO DE MEDIÇÃO DE BIOIMPEDÂNCIA

Diversos autores como Aroom et al. (2009), Cerávolo (2012), Do Amaral et al. (2011), Hussain (2012), Martinez-Nolasco et al. (2012) , Papezova (2003), Porto (2009), e Yang et al. (2006), mostram variações de topologia de circuitos para a medição de bioimpedância. A partir dessas topologias, pode-se definir os blocos funcionais de um sistema genérico de medição de bioimpedância na técnica de quatro eletrodos. A Figura 4 mostra o diagrama de blocos do sistema genérico de medição de bioimpedância obtido com a generalização das topologias apresentadas nos trabalhos dos autores citados.

Figura 4 - Sistema genérico de medição de bioimpedância



Uma fonte de corrente controlada por tensão “VCCS” recebe uma tensão de controle de um gerador de sinal produzindo a corrente padrão na frequência utilizada de acordo com a técnica de bioimpedância empregada. A saída da fonte é conectada ao corpo que está sendo examinado por dois eletrodos, um para inserir a corrente e outro para enviar a corrente à impedância padrão (Z_{SHUNT}).

Um segundo par de eletrodos é utilizado para conectar um amplificador de instrumentação ao corpo que está sendo examinado, permitindo a medição da diferença de potencial sobre o mesmo. Um segundo amplificador de instrumentação conectado sobre a impedância de *shunt*, mede a tensão resultante sobre a mesma, permitindo que seja determinada a corrente que está passando, uma vez que o valor da impedância de *shunt* é conhecido. Cada amplificador de instrumentação envia o sinal medido para um bloco condicionador de sinal, o qual eleva ou diminui a tensão, casa a impedância com os detectores de amplitude e de fase, e algumas vezes, filtram o sinal. Os detectores de amplitude e de fase geram dois sinais distintos, função da fase e da amplitude do sinal vindo do bloco condicionador de sinal.

Os sinais que representam a fase e a amplitude da tensão sobre o tecido sob exame, e da corrente sobre a impedância de *shunt*, são então enviados para o bloco de medição de bioimpedância e controle. Esse bloco calcula o valor equivalente da impedância com base na tensão e na corrente medidas, controla o gerador de sinal para manter o valor da corrente injetada no tecido dentro da frequência e amplitude adequadas, e envia os valores da resistividade e reatância capacitiva para o bloco de cálculo de correlação fisiológica. Este último, determina a quantidade total de água do corpo (TBW), a massa magra (FFM), e dependendo da técnica de bioimpedância para qual o equipamento foi projetado, calcula também a quantidade de água extracelular (ECW) e a quantidade de água intracelular (ICW).

1.1.5 ESPECIFICAÇÕES PARA O CASO DE ESTUDO

A aplicação de interesse para a pesquisa deste trabalho é uma aplicação de medição de bioimpedância complexa pela técnica de análise bioelétrica com frequência única (SFBIA). A corrente alternada a ser injetada no tecido sob exame deve ser senoidal e ter frequência fundamental de 50 kHz. O foco do trabalho se concentra no desenvolvimento dos blocos de detecção da amplitude e detecção da fase do sistema genérico de medição de bioimpedância mostrado na Figura 4. A Tabela 1 mostra as especificações e os valores mínimos para o desempenho desejado.

Tabela 1 - Especificações do caso de estudo

Parâmetro	Valor
Técnica de Bioimpedância	SFBIA – 50 kHz
Faixa de Impedância Típica	200 Ω a 800 Ω
Corrente injetada	1 mA
Resolução de Magnitude Mínima	1 Ω
Resolução Angular Mínima	1°
Resolução Angular Desejável	0,1°
Erro para Resolução de 1°	$\emptyset^\circ + \text{erro} < \emptyset^\circ + 1^\circ$
	$\emptyset^\circ - \text{erro} > \emptyset^\circ - 1^\circ$
Erro para Resolução de 0,1°	$\emptyset^\circ + \text{erro} < \emptyset^\circ + 0,1^\circ$
	$\emptyset^\circ - \text{erro} > \emptyset^\circ - 0,1^\circ$

A partir da definição da aplicação, e com base nessas premissas e nos valores característicos de bioimpedância reportados na literatura, foram então definidos os valores de especificações que determinam a performance mínima a ser obedecida pela topologia da aplicação estudada. Essas especificações apresentadas são utilizadas como referência de performance durante o processo de verificação do modelo teórico das técnicas estudadas e na análise de resultados das simulações para a escolhas das técnicas de medição a serem implementação em FPGA.

1.2 OBJETIVOS

Partindo-se da existência das ferramentas reconfiguráveis de *hardware* (FPGA), o objetivo dessa pesquisa é comparar técnicas existentes para a medição de impedância, e selecionar um conjunto delas, que se adequem à medição (das componentes complexas) de bioimpedância, de forma a garantir a resolução da medida especificada e utilizar a menor quantidade de recursos da pastilha de silício que seja factível.

1.3 OBJETIVOS ESPECÍFICOS

Dentre os objetivos específicos do presente trabalho, destacam-se:

- identificar os métodos de medição de impedância que separadamente ou em conjunto possam medir as componentes complexas de um sinal;
- simular os métodos de medição identificados para verificar o tipo de resposta e erro específico;
- selecionar duas ou mais técnicas de medição dentre as técnicas identificadas para a implementação em código VHDL, e realizar a simulação para verificação do atendimento as resoluções de medida especificadas;

- avaliar o desempenho das técnicas implementadas em código VHDL dentro do FPGA.

1.4 ORGANIZAÇÃO DO TRABALHO

No capítulo 2 são fornecidos os princípios teóricos de sete métodos de medição de diferença de fase amplamente empregados ao longo da última década: derivação da figura de Lissajous, ajuste de elipse, demodulação de quadratura, medição direta digital por contagem de pulsos, PWM (Modulação de Lagura de Pulso), comparação de transformada discreta de Fourier, e ajuste de senóide.

No capítulo 3 são apresentados os modelos construídos no Simulink para as simulações de cada um dos métodos, os detalhes necessários para execução da simulação, os resultados obtidos com as simulações e a análise desses resultados. Após a análise de resultados, são definidos os critérios para a escolha dentre os métodos simulados, daqueles que seriam uma boa opção para se seguir com a implementação do método em código VHDL, e a seleção dos métodos para implementação

No capítulo 4 são apresentados os métodos selecionados implementados em VHDL, bem como a análise das simulações executadas através da ferramenta System Generator e da co-simulação de *hardware*.

No capítulo 5 são apresentadas as conclusões obtidas com a pesquisa e as perspectivas de trabalhos.

No anexo A é apresentado o algoritmo de DFT Radix-4 escrito em *script* MATLAB utilizado durante os testes do método de DFT.

No apêndice A é mostrada o procedimento para a execução da análise de sensibilidade ao ruído para todos os métodos de detecção de fase estudados nesse trabalho bem como seus resultados.

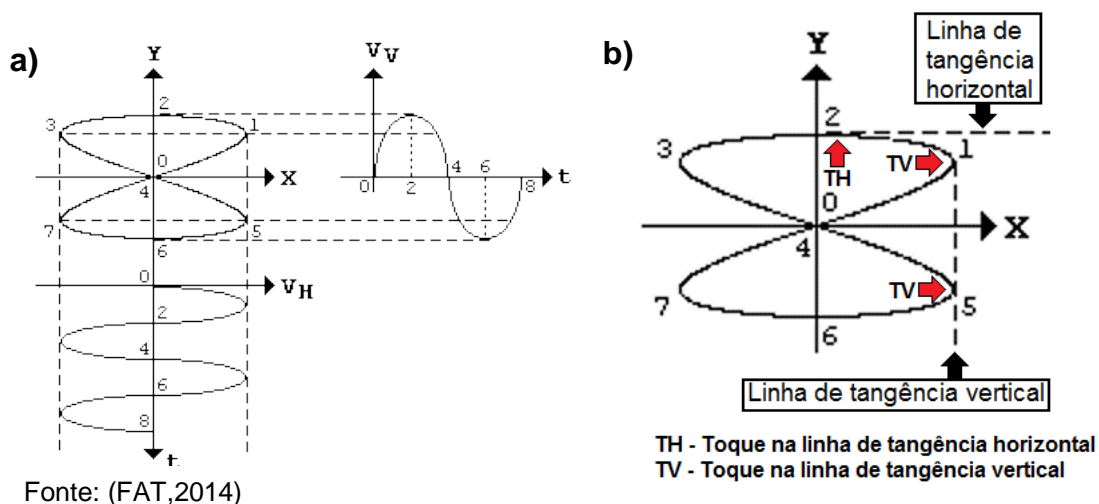
2. TÉCNICAS DE MEDIÇÃO DE DIFERENÇA DE FASE E MAGNITUDE

Este capítulo apresenta o embasamento teórico para sete métodos de medição de diferença de fase e magnitude. Quatro dos métodos permitem a determinação simultânea da relação de magnitude e diferença de fase dos sinais comparados, são eles: método do ajuste da elipse, método da demodulação de quadratura, método da transformada discreta de Fourier e método do ajuste da senóide. Os outros três métodos são específicos para a medição da diferença de fase; são eles: método da figura de Lissajous, método da medição direta digital (DDM – *Direct Digital Measurement*) e método da modulação de largura de pulsos (PWM – *Pulse Width Modulation*).

2.1 TÉCNICA DA FIGURA DE LISSAJOUS

A composição gráfica a partir dos valores de amplitude instantânea de dois movimentos ondulatórios, sendo um representado na horizontal e outro na vertical, resulta em uma figura denominada Figura de Lissajous, a qual é ilustrada na Figura 5. (FAT, 2014)

Figura 5 - a) Figura de Lissajous resultante da composição de dois sinais; b) Detalhe das linhas de tangência para determinação da relação de frequências.



A figura de Lissajous ilustrada na Figura 5a é resultado da composição de um sinal na vertical com frequência f , e um outro sinal na horizontal com frequência $2f$. Traçando-se uma linha tangente ao topo da figura de Lissajous (linha de tangência horizontal), e uma linha tangente à uma das laterais (linha de tangência vertical), pode-se estabelecer a relação entre as frequências dos dois sinais. Essa relação é obtida conforme o número de vezes que a Figura de Lissajous toca nas linhas de tangência horizontal e na vertical.

No exemplo da Figura 5b, a figura toca na linha de tangência horizontal uma vez e na linha de tangência vertical duas vezes. Portanto a relação de frequências será conforme indicada na equação (4). Para um caso genérico, a relação pode ser reescrita conforme a equação (5).

$$F_H = 2 \cdot F_V \therefore F_V/F_H = 1/2 \quad (4)$$

$$\frac{F_V}{F_H} = \frac{N_H}{N_V} \quad (5)$$

Onde:

F_V é a frequência do sinal vertical em hertz (ou rad/s).

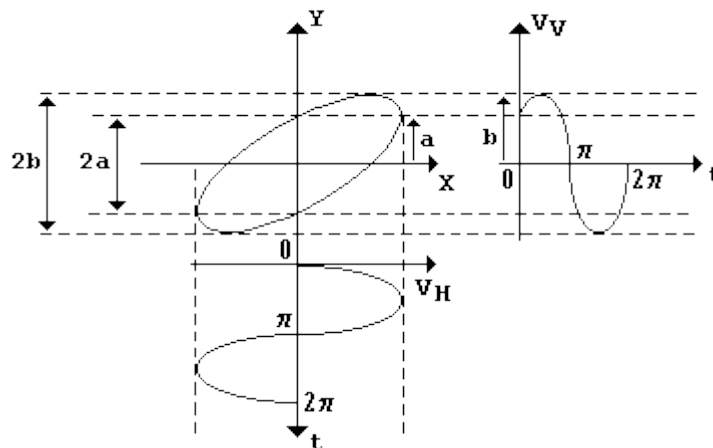
F_H é a frequência do sinal horizontal em hertz (ou rad/s).

N_H é o número de tangências na horizontal.

N_V é o número de tangências na vertical.

Se ambos os sinais vertical e horizontal forem sinais senoidais defasados e de mesma frequência, a Figura de Lissajous resultante será uma elipse, conforme é ilustrado na Figura 6.

Figura 6 - Figura de Lissajous para dois sinais senoidais de mesma frequência



Fonte: (FAT,2014)

Conforme mostrado na Figura 6, existe um parâmetro a , que representa a distância entre o centro da elipse e o ponto onde a elipse corta o eixo y , e um parâmetro b , que representa a distância entre o centro da elipse e o ponto máximo da figura de Lissajous. Para a determinação da defasagem entre os dois sinais, basta calcular-se o arco seno da razão de a por b . A origem dessa relação é obtida matematicamente da forma descrita a seguir.

Se o sinal horizontal for considerado como um sinal de referência, a fase desse sinal será a fase de referência para a determinação de defasagem. Considerando-se que o valor da fase de referência como 0° , a fase relativa do sinal vertical em relação ao sinal horizontal é $\Delta\varphi$. Assim, se ambos os sinais são sinais senoidais, o sinal vertical e o sinal horizontal podem ser descritos matematicamente pelas equações (6) e (7) respectivamente.

$$V_V(t) = V_{Vmax} \cdot \sin(2 \cdot \pi \cdot f \cdot t + \Delta\varphi) + C_V \quad (6)$$

$$V_H(t) = V_{Hmax} \cdot \sin(2 \cdot \pi \cdot f \cdot t) + C_H \quad (7)$$

Onde:

V_V é a amplitude instantânea do sinal vertical em volts.

V_H é a amplitude instantânea do sinal vertical em volts.

V_{Vmax} é a amplitude máxima do sinal vertical em volts.

V_{Hmax} é a amplitude máxima do sinal horizontal em volts.

f é a frequência dos sinais em hertz.

$\Delta\varphi$ é a fase relativa do sinal vertical em radianos.

t é o instante de tempo em segundos.

C_V é a tensão de *offset* CC do sinal vertical em volts.

C_H é a tensão de *offset* CC do sinal horizontal em volts.

Se a origem de um sistema cartesiano (x, y) for imposta no centro da elipse apresentada na Figura 6, o valor da amplitude máxima V_{Vmax} do sinal vertical é equivalente à distância b entre o centro da elipse e o ponto máximo da Figura de Lissajous. Essa relação pode ser expressa conforme a equação (8).

$$V_{Vmax} = b \quad (8)$$

Onde:

V_{Vmax} é a amplitude máxima do sinal vertical em volts.

b é a distância, em volts, entre o centro da elipse e o ponto máximo da Figura de Lissajous.

Ainda, conforme mostrado na Figura 6, o parâmetro a é equivalente à amplitude do sinal vertical tanto no instante de tempo 0 segundos quanto no instante de tempo onde $\omega \cdot t = \pi$ radianos. Essa periodicidade se deve ao fato de que a elipse corta o eixo y no mesmo instante em que o valor do sinal horizontal é zero. Então, para o instante de tempo $t = 0$ s, a amplitude instantânea do sinal vertical $V_V(t)$ é $V_V(t) = V_V(0) = a$. Substituindo-se os valores de V_{Vmax} , $V_V(0)$, e o instante de tempo $t = 0$ s na equação (6), obtém-se a equação (9).

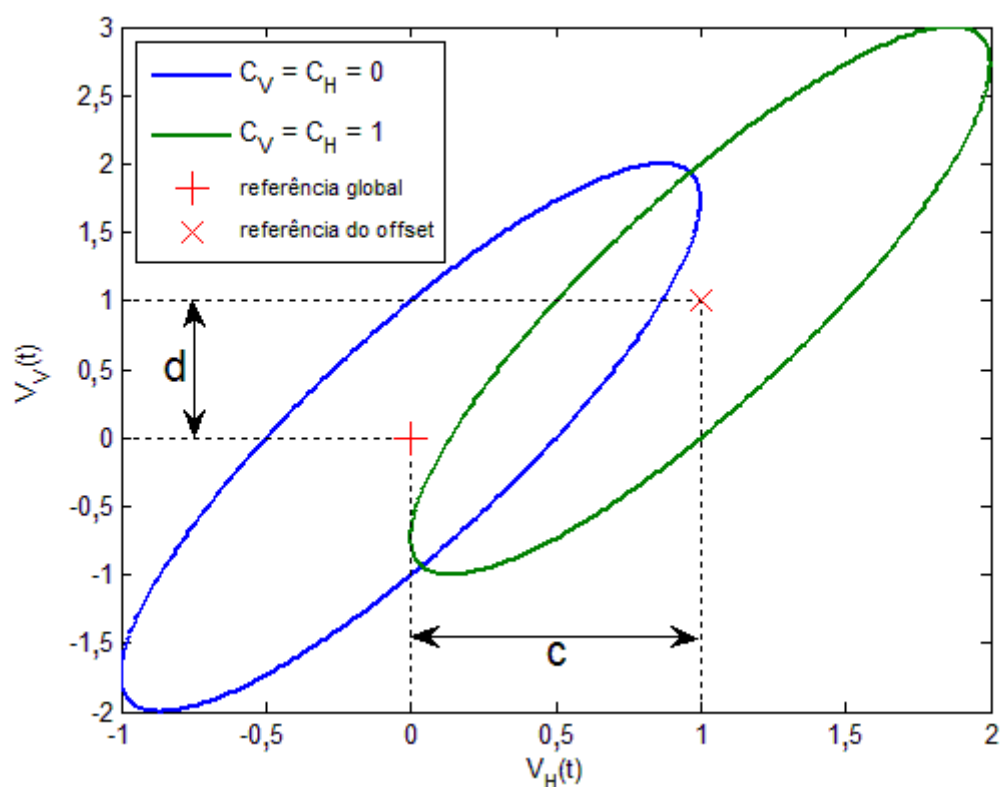
$$a = b \cdot \text{sen}(2 \cdot \pi \cdot f \cdot 0 + \Delta\varphi) \therefore a = b \cdot \text{sen}(\Delta\varphi) \quad (9)$$

Finalmente, isolando-se o termo $\Delta\varphi$ na equação (9), obtém-se a equação (10), a qual permite o cálculo da diferença de fase entre os dois sinais.

$$\Delta\varphi = \arcsen\left(\frac{a}{b}\right) \quad (10)$$

Conclui-se então que para se determinar a defasagem através da elipse obtida na figura de Lissajous para dois sinais de mesma frequência, basta obter-se os valores de a e b , onde a representa a distância entre o centro da elipse e o ponto onde esta corta o eixo y , e b representa a distância entre o centro da elipse e o ponto máximo da figura. No caso de existir *offset* de tensão CC nos sinais vertical e/ou horizontal, irá se verificar o deslocamento do centro da elipse em relação à origem do centro global de referência. A Figura 7 ilustra esse deslocamento.

Figura 7- Comparação das elipses de um sinal com *offset* e outro sem *offset*



Na Figura 7 estão plotadas as curvas de duas composições: uma na qual os sinais horizontal e vertical tem *offset* zero ($C_V = C_H = 0$) e outra na qual ambos tem *offset* um ($C_V = C_H = 1$). A frequência dos sinais utilizados é de 50 kHz, a

amplitude $V_{Vmax} = 2$, a amplitude $V_{Hmax} = 1$ e os pares ordenados $(V_H(t), V_V(t))$ foram obtidos com 360 passos para o intervalo de tempo $0 \leq t \leq 1/(50 \times 10^3)$.

Conforme ilustrado, observa-se que se a distância vertical e a distância horizontal entre o centro da elipse com *offset* e o centro global de referência forem calculadas, é possível se terminar o *offset* do sinal vertical e o *offset* do sinal horizontal. Deste modo, sendo c a distância horizontal entre o eixo global de referência e o centro da elipse com *offset*, e d a distância vertical, as equações (11) e (12) calculam o *offset* do sinal horizontal e o *offset* do sinal vertical respectivamente.

$$c = C_H \quad (11)$$

$$d = C_V \quad (12)$$

A partir da técnica gráfica da Figura de Lissajous é possível derivar um método matemático de medição da diferença de fase entre um sinal medido e um sinal de referência sem a necessidade da visualização gráfica da elipse. Tendo como premissa que o sinal horizontal é um sinal de referência cuja amplitude $V_H = 0$, fase φ_H e *offset* C_H são conhecidos, o sinal vertical a ser medido possui uma diferença de fase $\Delta\varphi$, amplitude de pico V_{Vpico} , amplitude de vale V_{Vvale} , e *offset* C_V . Basta então definir como se obter os parâmetros a e b .

Para a determinação do parâmetro a leva-se em consideração o instante no qual o sinal de referência assume valor zero ($V_H = 0$). Nesse instante guarda-se uma amostra V_a do valor do sinal medido. Esse valor corresponde ao valor $a + C_V$. Logo, o parâmetro a pode ser estimado através da equação (13).

$$a = V_a - C_V \quad (13)$$

Para a determinação do parâmetro b , devem-se detectar o valor de pico positivo do sinal medido e o valor de pico negativo (valor de vale). Uma vez que estes dois valores são conhecidos, o valor do *offset* do sinal medido pode ser calculado através da equação (14).

$$C_V = \frac{V_{Vpico_positivo} - V_{Vpico_negativo}}{2} \quad (14)$$

Se o sinal medido não possuir *offset*, de acordo com a equação (8), o valor do parâmetro b será igual ao valor da amplitude do sinal medido. Se o sinal possuir *offset*, o valor da amplitude V_{Vmax} deve primeiramente ser calculado para que se determine em seguida o valor do parâmetro b . Assim, subtraindo-se do valor de pico medido V_{Vpico} o valor do *offset* vertical calculado C_V , determina-se o parâmetro b através da equação (15).

$$b = V_{Vmax} = V_{Vpico} - C_V \quad (15)$$

Por fim, basta utilizar a equação (10) para se obter o valor da diferença de fase do sinal medido.

O método derivado da técnica da figura de Lissajous se mostra matematicamente bastante simples, sendo basicamente composto por operações de soma, subtração e divisão. A operação mais complexa é a operação de arco-seno. Apesar desse método se mostrar bastante vantajoso pela sua simplicidade, por ser um método de medição direta, ele é bastante suscetível a ruído. Qualquer perturbação que cause erro no momento de amostrar o sinal medido quando o sinal de referência assume valor zero pode causar erros bastante significativos. Dessa forma, uma alternativa direta ao método derivado da técnica da figura de Lissajous é o método do ajuste da elipse, para o caso específico de sinais de mesma frequência.

2.2 TÉCNICA DO AJUSTE DA ELIPSE

Conforme mostrado na técnica da figura de Lissajous, para dois sinais senoidais, pode-se plotar uma figura na qual os pares ordenados são formados por um valor no eixo x , pertencente ao sinal senoidal 1 e um valor no eixo y pertencente ao sinal senoidal 2. Caso os dois sinais senoidais possuam a mesma frequência, a curva resultante será uma reta quando as mesmas estão em fase ou oposição de fases (180°), ou então uma elipse, para outro valor de diferença

de fase. Com base nesse princípio, é possível se utilizar do método dos mínimos quadrados, para, a partir de um conjunto de pares ordenados obtidos de dois sinais senoidais amostrados, ajustar-se uma curva elíptica baseada na equação cônica da elipse. A partir dos coeficientes da equação da elipse é possível então se determinar a diferença de fase, amplitudes das duas ondas senoidais e *offsets*.

Um dos algoritmos mais citados para o ajuste de uma elipse a partir de uma nuvem de pontos é o algoritmo proposto por Halir e Flusser (1998). Nele, parte-se do fato de que uma elipse é um caso especial da equação cônica mostrada na equação (16), e que se ela obedecer a restrição mostrada na equação (17), a cônica se transformará em uma elipse

$$F(x, y) = a \cdot x^2 + b \cdot x \cdot y + c \cdot y^2 + d \cdot x + e \cdot y + g = 0 \quad (16)$$

$$b^2 - 4 \cdot a \cdot c < 0 \quad (17)$$

Nas equações (16) e (17), a, b, c, d, e, g são os coeficientes da elipse e (x, y) são as coordenadas dos pontos que descrevem a elipse. O polinômio $F(x, y)$ é denominado de distância algébrica dos pontos (x, y) . Utilizando-se a notação vetorial, a equação (16) pode ser escrita na forma da equação (20), através dos vetores representados nas equações (18) e (19).

$$\vec{a} = [a, b, c, d, e, g]^T \quad (18)$$

$$\vec{x} = [x^2, xy, y^2, x, y, 1] \quad (19)$$

$$F_a(\vec{x}) = \vec{x} \cdot \vec{a} \quad (20)$$

O ajuste de uma cônica geral a um conjunto de pontos pode ser aproximado minimizando-se a soma do quadrado das distâncias algébricas dos pontos da cônica. Se a cônica é representada pelos coeficientes do vetor \vec{a} e o conjunto dos pontos é expresso como: $(x_i, y_i), i = 1 \cdots N$, o problema pode ser descrito de acordo com a equação (21).

$$\min_{\vec{a}} \sum_{i=1}^N F(x_i, y_i)^2 = \min_{\vec{a}} \sum_{i=1}^N F_{\vec{a}}(\vec{x}_i)^2 = \min_{\vec{a}} \sum_{i=1}^N (\vec{x}_i \cdot \vec{a})^2 \quad (21)$$

O problema expresso pela equação (21) pode ser resolvido pela técnica do método dos mínimos quadrados, porém, a solução pode não ser necessariamente uma elipse. Para garantir que isso não ocorra, a restrição da equação (17) deve ser considerada. A restrição é imposta reescrevendo-a da forma de uma inequação para a forma de uma igualdade, conforme mostra a equação (22).

$$4 \cdot a \cdot c - b^2 = 1 \quad (22)$$

O processo comum de resolução do problema de minimização é realizado através do processo dos mínimos quadrados com restrições quadráticas, e pela aplicação dos multiplicadores de Lagrange, o que resultará em seis autovetores cujas componentes são os coeficientes da equação da elipse. O autovetor que corresponde ao autovalor mínimo positivo é o conjunto de coeficientes que melhor representa a elipse ajustada.

Apesar desse procedimento ser matematicamente correto, o cálculo numérico dos autovalores é numericamente instável e pode produzir resultados errados quando as matrizes utilizadas no processo matemático se aproximam de uma matriz singular. Do ponto de vista da localização da solução ótima, o autovalor ótimo pode ser um número negativo muito pequeno, aproximando-se de zero, ao invés do mínimo valor positivo. Para evitar-se esses problemas, Halir e Flusser (1998) propuseram uma alteração no método de cálculo da solução ótima. O método de cálculo consiste na resolução do sistema de equações composto pelas equações (23) a (24).

$$\mathbf{M} \cdot \vec{a}_1 = \lambda \cdot \vec{a}_1 \quad (25)$$

$$\vec{a}_1^T \cdot \mathbf{C}_1 \cdot \vec{a}_1 = 1 \quad (26)$$

$$\vec{a}_2 = -\mathbf{S}_3^{-1} \cdot \mathbf{S}_2^T \cdot \vec{a}_1 \quad (27)$$

$$\vec{a} = \begin{bmatrix} \vec{a_1} \\ \vec{a_2} \end{bmatrix} \quad (28)$$

$$\mathbf{M} = \mathbf{C}_1^{-1} \cdot (\mathbf{S}_1 - \mathbf{S}_2 \cdot \mathbf{S}_3^{-1} \cdot \mathbf{S}_2^T) \cdot \vec{a_1} \quad (29)$$

Nesse sistema, \vec{a} é o autovetor, λ é o multiplicador de Lagrange, \mathbf{M} é a matriz de dispersão reduzida, com dimensões de 3×3 , \mathbf{S}_1 , \mathbf{S}_2 e \mathbf{S}_3 são sub-matrizes de uma matriz de dispersão e \mathbf{C}_1 é uma sub-matriz de uma matriz de restrição que define o problema de mínimos quadrados com restrições quadráticas. Considerando-se que cada ponto amostrado forma um par ordenado (x_i, y_i) , onde x_i é o valor amostrado para uma primeira onda e y_i é o valor amostrado para uma segunda, o conjunto de pontos obtidos para um total de N amostras forma duas sub-matrizes de *design* denotadas por \mathbf{D}_1 e \mathbf{D}_2 .

As duas sub-matrizes de *design* \mathbf{D}_1 e \mathbf{D}_2 , cujas descrições matemáticas são apresentada nas equações 28 e 29 respectivamente, são utilizadas para se determinar as sub-matrizes de dispersão \mathbf{S}_1 , \mathbf{S}_2 e \mathbf{S}_3 conforme as relações matemáticas expressas pelas equações (30) a (32). A sub-matriz de dispersão \mathbf{C}_1 é uma matriz de constante e é expressa pela equação (33). As componentes do autovetor \vec{a} relacionam-se com os coeficientes da elipse conforme as equações (34) e (35).

$$\mathbf{D}_1 = \begin{bmatrix} x_1^2 & x_1 \cdot y_1 & y_1^2 \\ \vdots & \vdots & \vdots \\ x_i^2 & x_i \cdot y_i & y_i^2 \\ \vdots & \vdots & \vdots \\ x_N^2 & x_N \cdot y_N & y_N^2 \end{bmatrix} \quad (30)$$

$$\mathbf{D}_2 = \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_i & y_i & 1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & 1 \end{bmatrix} \quad (31)$$

$$\mathbf{S}_1 = \mathbf{D}_1^T \cdot \mathbf{D}_1 \quad (32)$$

$$S_2 = D_1^T \cdot D_2 \quad (33)$$

$$S_3 = D_2^T \cdot D_2 \quad (34)$$

$$C_1 = \begin{bmatrix} 0 & 0 & 2 \\ 0 & -1 & 0 \\ 2 & 0 & 0 \end{bmatrix} \quad (35)$$

$$\vec{a}_1 = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (36)$$

$$\vec{a}_2 = \begin{bmatrix} d \\ e \\ g \end{bmatrix} \quad (37)$$

Uma vez que a equação (38) é a equação que impõe o problema de autovetores e autovalores, é necessário encontrar apenas o autovetor \vec{a}_1 apropriado para a matriz M . A matriz possui apenas três autovalores reais, tipicamente dois negativos e um positivo. Para se efetuar a resolução do sistema de equações (39) a (40) de forma numérica, Halir e Flusser (1998) propuseram o algoritmo na forma de script MATLAB descrito na Figura 8.

Figura 8 - Algoritmo MATLAB de ajuste de elipse proposto por Halir e Flusser.

```

1  function a = fit_ellipse(x,y)
2  -   D1 = [x.^2, x.*y, y.^2]; % parte quadrática da matriz de design
3  -   D2 = [x, y, ones(size(x))]; % parte linear da matriz de design
4  -   S1 = D1' * D1; % parte quadrática da matriz de dispersão
5  -   S2 = D1' * D2; % parte combinada da matriz de dispersão
6  -   S3 = D2' * D2; % parte linear da matriz de dispersão
7  -   T = - inv(S3) * S2'; % para se obter o vetor a2 à parte do vetor a1
8  -   M = S1 + S2 * T; % matriz de dispersão reduzida
9  -   M = [M(3,:)./2; -M(2,:); M(1,:)./2]; % pré-multiplicação pelo inversa de C1
10 -   [evec, eval] = eig(M); %ok<NASGU> resolve o auto-sistema(valores, vetores)
11 -   cond = 4*evec(1,:).*evec(3,:)-evec(2,:).^2; %avalia a'Ca
12 -   a1=evec(:, find(cond>0)); %ok<FNDSB> auto-vetor para o min auto-valor pos.
13 -   a = [a1; T * a1]; %coeficientes da elipse

```

Fonte: (HALIR; FLUSSER, 1998)

No algoritmo, dois artifícios são utilizados para melhorar a estabilidade numérica. O primeiro é que devido a estrutura especial da matriz C_1 (35), a pré-multiplicação por C_1^{-1} que é necessária na equação **Error! Reference source not found.** é executada na linha 9 do algoritmo. O segundo artifício resolve o problema da localização do mínimo valor não negativo do autovalor da matriz de dispersão reduzida M , quando esta se aproxima de uma matriz singular. Ao invés de se localizar o menor valor não negativo de autovalor, compara-se a condição da equação (41) para todos os autovetores da matriz M (linha 11 da Figura 8). Verifica-se que existe apenas um único autovetor que gera um valor positivo, valor esse, correspondente a ótima solução do problema de ajuste. O autovetor é localizado na linha 12. E finalmente na linha 13, o resto dos coeficientes da elipse são calculados fornecendo a solução completa.

Uma vez que os coeficientes da elipse tenham sido determinados, as relações de diferença de fase, amplitude e componente CC podem ser determinadas. As equações para obtenção destes valores são mostradas por Ramos et al. (2009) na implementação de um medidor de bioimpedância em DSP. Supondo-se que os dois sinais amostrados para compor os pontos da elipse são representados pelas equações (42) e (43), a equação da elipse (equação (16)), pode ser reescrita na forma da equação (44).

$$v_1(t) = D_1 \cdot \text{sen}(2 \cdot \pi \cdot f \cdot t + \varphi_1) + C_1 \quad (42)$$

$$v_2(t) = D_2 \cdot \text{sen}(2 \cdot \pi \cdot f \cdot t + \varphi_2) + C_2 \quad (43)$$

$$F(v_1, v_2) = a \cdot v_1^2 + b \cdot v_1 \cdot v_2 + c \cdot v_2^2 + d \cdot v_1 + e \cdot v_2 + g = 0 \quad (44)$$

A razão de amplitudes pode ser estimada através da equação (45). Já as componentes de *offset* CC podem ser obtidas com as equações (46) e (47), e a diferença de fase pode ser obtida com a equação (48).

$$\frac{D_2}{D_1} = \sqrt{\frac{a}{c}} \quad (45)$$

$$C_1 = \frac{2 \cdot c \cdot d - b \cdot e}{b^2 - 4 \cdot a \cdot c} \quad (46)$$

$$C_2 = \frac{2 \cdot a \cdot e - b \cdot d}{b^2 - 4 \cdot a \cdot c} \quad (47)$$

$$\cos(\Delta\varphi) = -\frac{\text{sign}(a) \cdot b}{2 \cdot \sqrt{a \cdot c}} \quad (48)$$

Para determinar o sinal da diferença de fase é necessário implementar um sistema de voto que estime a direção de rotação da elipse. Isso é feito medindo-se o sinal do ângulo de rotação em relação ao baricentro dos pontos amostrados. As coordenadas do baricentro de uma elipse são atualizadas depois de cada segmento de dado ter sido adquirido. Para o cálculo do baricentro são utilizados os valores dos elementos S_{v_1} e S_{v_2} que foram calculados para se compor a matriz **S3** (equação (34)). Sendo $N_{acquired}$ o número de amostras obtidas até o momento, as equações (49) e (50) determinam as coordenadas do baricentro.

$$x_N = \frac{S_{v_1}}{N_{acquired}} \quad (49)$$

$$y_N = \frac{S_{v_2}}{N_{acquired}} \quad (50)$$

Este procedimento é também adequado para situações com componentes CC, uma vez que ele se aproxima dinamicamente do centro da elipse. Cada par de amostra (uma amostra de v_1 e outra amostra de v_2) define um número complexo no plano de Argand (também conhecido como plano complexo) centrado em x_N e y_N . A diferença de sinal entre ângulos consecutivos, mostrada na equação (51), define o voto de cada par amostrado.

$$\delta_n|_{n=1, \dots, N-1} = \text{sign}\{\arg[v_{1,n+1} - x_N + j \cdot (v_{2,n+1} - y_N)] - \arg[v_{1,n} - x_N + j \cdot (v_{2,n} - y_N)]\} \quad (51)$$

No geral, o número de votos que estimam corretamente a direção de rotação da elipse é sempre muito maior do que o número de votos que estimam de forma errada, mesmo em situações de baixo SNR (razão sinal ruído) ou quando a razão frequência de amostragem para frequência do sinal é alta. Isso ocorre, porque, já que as amostras criam uma elipse no plano Argrand, há uma tendência presente no sinal de ângulos consecutivos, a qual permite a correção da direção de rotação da elipse. Também, cada amostra contribui apenas com um único voto para o resultado final.

Utilizando-se os votos obtidos com a equação (51), o sinal estimado da diferença de fase $\Delta\varphi$ é obtido com a equação (52). Este método para determinação da direção de rotação da elipse funciona apenas se o critério de Nyquist para a frequência de amostragem estiver sendo cumprido (a frequência de amostragem deve ser pelo menos duas vezes a frequência do sinal sendo amostrado).

$$\text{sign}(\Delta\varphi) = -\text{sign}\left(\sum_{n=1}^{N-1} \delta_n\right) \quad (52)$$

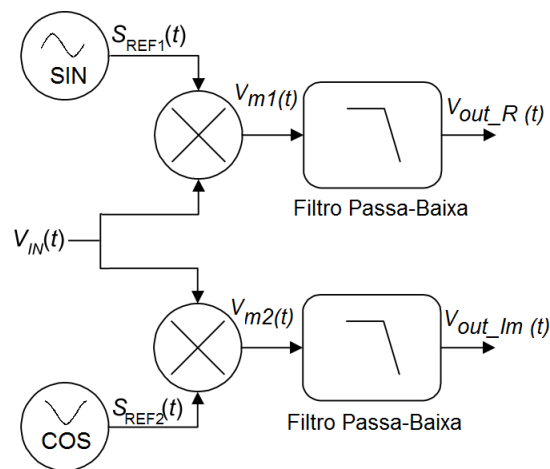
A técnica do ajuste da elipse se mostra bastante complexa matematicamente. A vantagem de se utilizar esse método é a capacidade de se compensar as perturbações oriundas de ruído pelo ajuste da elipse através do método dos mínimos quadrados. Segundo a aplicação da técnica do ajuste da elipse mostrada por Ramos et al. (2009), a relação sinal ruído obtida foi sempre melhor que 60 dB, resultando em um erro médio de estimativa de fase equivalente a $2,6 \times 10^{-6}$ graus.

2.3 TÉCNICA DO DEMODULADOR DE QUADRATURA

De acordo com Yang et al. (2006) o método de demodulação de quadratura também conhecido como PSD (*phase sensitive detector*) se tornou uma solução bastante popular para se determinar a diferença de fase. O demodulador de quadratura pode ser implementado tanto com circuito analógico

quanto digital. Uma proposta de um demodulador de quadratura digital é apresentada por Martinez-Nolasco et al. (2012), para uma aplicação em FPGA voltada para a medição de bioimpedância para frequências de até 10 MHz. Independentemente do demodulador de quadratura ser implementado de forma analógica ou de forma digital, o princípio de funcionamento é o mesmo. A Figura 9 mostra o diagrama em blocos de um demodulador de quadratura.

Figura 9 - Diagrama em blocos de um demodulador de quadratura



Adaptado de: (MARTINEZ-NOLASCO et al., 2012)

O princípio de funcionamento de um demodulador de quadratura se baseia em misturar um sinal de entrada $V_{IN}(t)$ que será demodulado em dois misturadores distintos. No primeiro misturador o sinal $V_{IN}(t)$ é misturado com um sinal de referência $S_{REF1}(t)$, normalmente senoidal. No segundo misturador, o sinal $V_{IN}(t)$ é misturado com um sinal de referência $S_{REF2}(t)$ defasado 90° em relação ao sinal de referência $S_{REF1}(t)$ utilizado no primeiro misturador. A saída de cada misturador é então aplicada em um filtro passa-baixa, o qual remove as componentes de frequência superiores à frequência de corte do filtro. Para o filtro passa-baixa do misturador com sinal de referência $S_{REF2}(t)$ defasado em 90° , obtém-se na saída do filtro um valor de tensão correspondente à componente cartesiana imaginária do sinal de entrada $V_{IN}(t)$. Para o filtro passa-baixa do misturador com sinal de referência $S_{REF1}(t)$ não defasado, obtém-se na saída do

filtro um valor de tensão correspondente à componente cartesiana real do sinal de entrada $V_{IN}(t)$. Na Figura 9 utiliza-se um sinal de referência senoidal com fase 0° e um sinal de referência cosseínodal com fase 0° . Este último, é equivalente a um sinal senoidal defasado em 90° em relação ao sinal senoidal de fase 0° .

Para se equacionar o demodulador de quadratura, considera-se inicialmente que o sinal de entrada $V_{IN}(t)$ é um sinal expresso na forma da equação (53), e os sinais de referência $S_{REF1}(t)$ e $S_{REF2}(t)$ utilizados, são expressos na forma das equações (54) e (55) respectivamente.

$$V_{IN}(t) = A \cdot \text{sen}(\omega \cdot t + \varphi_1) \quad (53)$$

$$S_{REF1}(t) = B \cdot \text{sen}(\omega \cdot t) \quad (54)$$

$$S_{REF2}(t) = B \cdot \cos(\omega \cdot t) \quad (55)$$

Uma vez que o misturador 1 mistura o sinal $V_{IN}(t)$ com o sinal $S_{REF1}(t)$, obtém-se na saída do misturador, um sinal V_{m1} resultado da relação mostrada na equação (56).

$$V_{m1}(t) = V_{IN}(t) \cdot S_{REF1}(t) \quad (56)$$

Substituindo-se as equações (53) e (54) na equação (56), obtém-se a equação (57).

$$V_{m1}(t) = A \cdot \text{sen}(\omega \cdot t + \varphi_1) \cdot B \cdot \text{sen}(\omega \cdot t) \quad (57)$$

Aplicando-se a relação trigonométrica da equação (58) na equação (57), obtém-se a equação (59).

$$\text{sen } x \cdot \text{sen } y = \frac{1}{2} \cdot [\cos(x - y) - \cos(x + y)] \quad (58)$$

$$V_{m1}(t) = \frac{A \cdot B}{2} \cdot [\cos(\omega \cdot t + \varphi_1 - \omega \cdot t) - \cos(\omega \cdot t + \varphi_1 + \omega \cdot t)] \quad (59)$$

Desenvolvendo-se a equação (59) obtém-se a equação (60).

$$V_{m1}(t) = \frac{A \cdot B}{2} \cdot [\cos(\varphi_1) - \cos(2 \cdot \omega \cdot t + \varphi_1)] \quad (60)$$

De forma análoga, o misturador 2 mistura o sinal de entrada $V_{IN}(t)$ com o sinal de referência $S_{REF2}(t)$, obtém-se na saída do misturador um sinal $V_{m2}(t)$, resultado da relação mostrada na equação (61).

$$V_{m2}(t) = V_{IN}(t) \cdot S_{REF2}(t) \quad (61)$$

Substituindo-se as equações (53) e (55) na equação (61), obtém-se a equação (62).

$$V_{m2}(t) = A \cdot \sin(\omega \cdot t + \varphi_1) \cdot B \cdot \cos(\omega \cdot t) \quad (62)$$

Aplicando-se a relação trigonométrica da equação (63) na equação (62), obtém-se a equação (64).

$$\cos x \cdot \sin y = \frac{1}{2} \cdot [\sin(x + y) - \sin(x - y)] \quad (63)$$

$$V_{m2}(t) = \frac{A \cdot B}{2} \cdot [\sin(\omega \cdot t + \omega \cdot t + \varphi_1) - \sin(\omega \cdot t - \omega \cdot t - \varphi_1)] \quad (64)$$

Desenvolvendo-se a equação (64) obtém-se a equação (65).

$$V_{m2}(t) = \frac{A \cdot B}{2} \cdot [\sin(2 \cdot \omega \cdot t + \varphi_1) - \sin(-\varphi_1)] \quad (65)$$

Sabendo-se que a relação trigonométrica da equação (66) é válida, a equação (65) pode ser reescrita na forma da equação (67).

$$\sin y = -\sin(-y) \quad (66)$$

$$V_{m2}(t) = \frac{A \cdot B}{2} \cdot [\sin(\varphi_1) + \sin(2 \cdot \omega \cdot t + \varphi_1)] \quad (67)$$

Ambos os sinais $V_{m1}(t)$ e $V_{m2}(t)$ são aplicados em filtros passa-baixa, os quais removem as componentes de frequência superiores à frequência de corte

do filtro. Selecionando-se a combinação correta de ordem do filtro passa baixa e de frequência de corte, e ainda, sendo que a frequência de corte deve ser próxima da frequência do sinal de referência utilizado, se a frequência angular do sinal de referência é ω , as componentes que tem frequência angular $2 \cdot \omega$ nas equações (60) e (67) serão removidas. Deste modo, a saída dos filtros passa baixa do demodulador de quadratura reduzem as equações (60) e (67) para as equações (68) e (69) respectivamente.

$$V_{out_R} \approx \frac{A \cdot B}{2} \cdot \cos(\varphi_1) \quad (68)$$

$$V_{out_Im} \approx \frac{A \cdot B}{2} \cdot \sin(\varphi_1) \quad (69)$$

As equações (68) e (69) representam coordenadas cartesianas, que quando convertidas para a forma polar, resultam nas equações (70) e (71).

$$\varphi_1 = \tan^{-1} \left(\frac{V_{out_Im}}{V_{out_R}} \right) \quad (70)$$

$$R = \sqrt{V_{out_Re}^2 + V_{out_Im}^2} = \frac{A \cdot B}{2} \quad (71)$$

A partir da descrição da saída dos filtros na forma polar verifica-se que a equação (70) corresponde à fase do sinal demodulado e que a equação (71) corresponde à metade da multiplicação entre a amplitude do sinal demodulado e a amplitude do sinal de referência. Utilizando-se um sinal de referência com amplitude igual a um, a equação (71) resultará na metade da amplitude do sinal demodulado.

Uma grande desvantagem do método do demodulador de quadratura é o possível descasamento de fase entre os sinais de referência. A implementação digital do demodulador pode diminuir consideravelmente esse problema uma vez que ambos os sinais podem ser gerados simultaneamente a partir de uma tabela e o processamento do sinal de forma digital ocorre de forma sincronizada a cada pulso de *clock* do sistema.

2.4 TÉCNICA DA MEDIÇÃO DIRETA DIGITAL

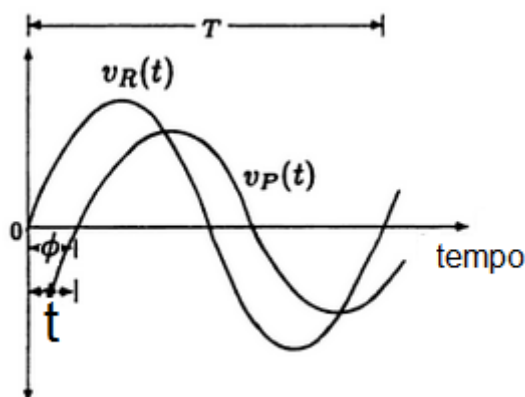
Na técnica de medição direta digital para bioimpedância, normalmente utiliza-se um medidor digital para diferença de fase e um outro medidor digital para amplitude dos sinais. Uma proposta de um medidor digital de bioimpedância que mede diferença de fase e a razão de amplitudes entre o sinal medido e o sinal de referência é apresentada por Rathore (2011). Nela, um conversor analógico-digital de dupla rampa é utilizado para a medição das razões de amplitudes e um contador de pulsos é utilizado para a medição de fase.

De acordo com Rathore (2003), a medição de fase de forma digital pode ser feita através da medida de tempo ou da medição direta de fase, sendo que na medição de fase através da medida de tempo existe a desvantagem de ser necessária a execução de cálculos aritméticos para se determinar qual é a fase.

A Figura 10 mostra a relação de fase entre dois sinais a serem comparados, sendo, $V_R(t)$ o sinal de referência e $V_P(t)$ o sinal defasado a ser medido. A relação de fase entre os dois sinais pode ser expressa em termos do período T , do sinal de referência, e do tempo de atraso t do sinal defasado, conforme mostra a equação (72).

$$\phi = 360 \cdot (t/T) \quad (72)$$

Figura 10 - Relação de fase entre dois sinais



Fonte: (RATHORE, 2003)

No método de medição de fase por medida de tempo, são medidos ambos o período T e o tempo de atraso t . Esses valores são então aplicados a equação (72), e o valor da fase em graus é determinado. Tanto a medição do tempo de atraso quanto a medição do período são feitas através da contagem de pulsos de *clock* cuja frequência fc é conhecida. Deste modo, para um determinado intervalo de tempo t , são contados N pulsos de *clock* cujo período é de $1/fc$, permitindo a relação mostrada na equação (73).

$$t = N \cdot (1/fc) \quad (73)$$

Substituindo-se a equação (73) na equação (72), obtém-se a equação (74), na qual o número de pulsos contados varia de acordo com a frequência.

$$N = \left(\frac{fc}{360 \cdot f} \right) \cdot \phi \quad (74)$$

A partir da equação (74), verifica-se que o instrumento pode medir diretamente o ângulo de fase se a razão (fc/f) for escolhida com um valor na forma $3,6 \times 10^n$. Nesse caso, o número de pulsos medidos e o ângulo de fase se relacionam de acordo com a equação (75).

$$N = 10^{n-2} \cdot \phi \quad (75)$$

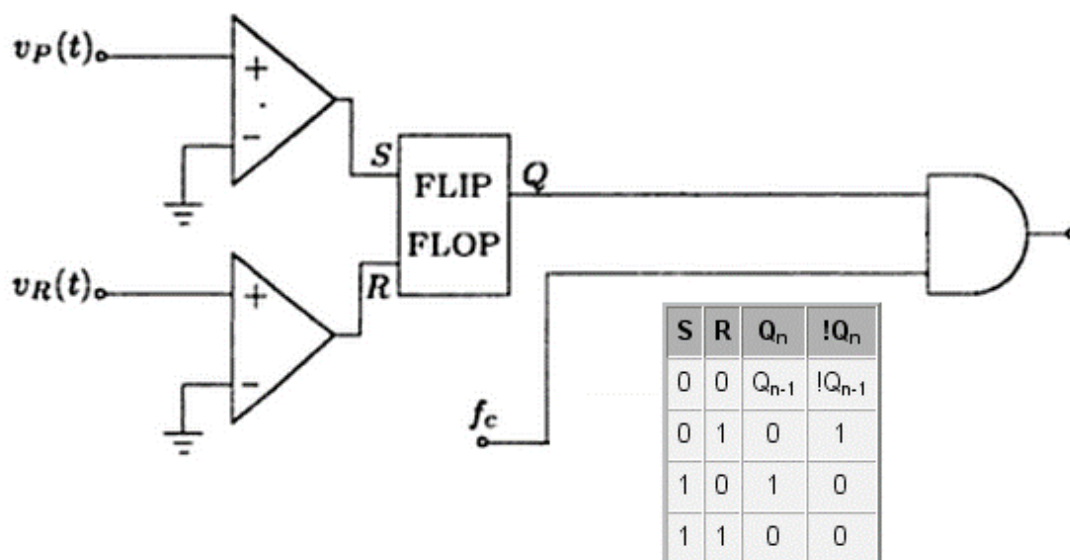
Supondo-se uma defasagem de um grau, utiliza-se a equação (75) para se determinar o valor de n para uma resolução angular desejada (expressa em um número de pulsos N que equivalem a um grau), e a equação (76) para se determinar a frequência de *clock*, que para a frequência do sinal de referência, resulta na resolução angular desejada. A Tabela 2 exemplifica o uso das equações (73) e (74) para o cálculo da frequência de *clock* para diversas resoluções angulares para um sinal de referência com frequência de 100 kHz. A frequência de 100 kHz foi escolhida para facilidade de cálculo.

$$fc = 3,6 \times 10^n \cdot f \quad (76)$$

Tabela 2 - Exemplo de cálculo de frequência de *clock* dos pulsos de contagem

Resolução (°)	\emptyset (°)	$N = (1/\text{resolução}) * \emptyset$	n [eq (75)]	f_c [eq (76)]
0,001	1	1000	5	36 GHz
0,010	1	100	4	3,6 GHz
0,100	1	10	3	360 MHz
1,000	1	1	2	36 MHz

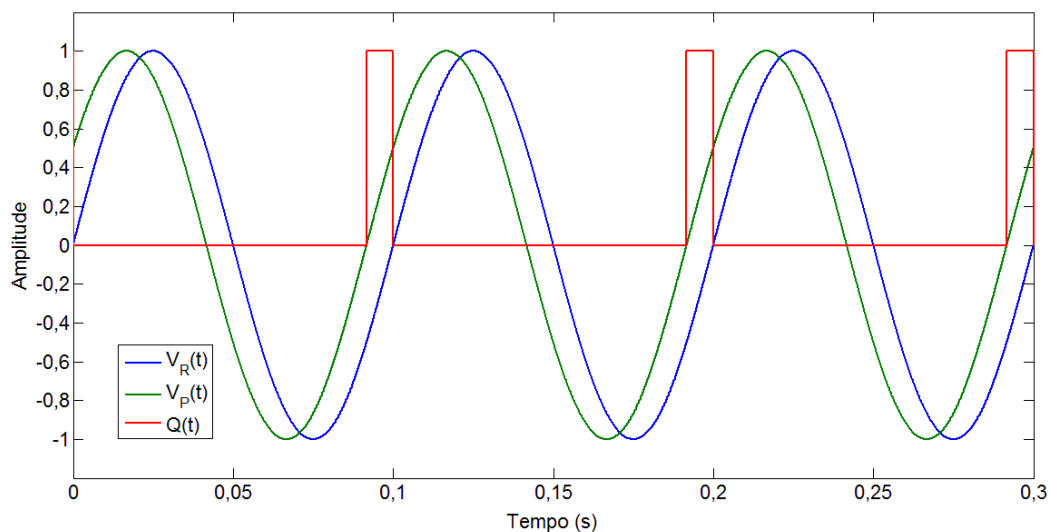
Segundo Rathore (2003), a implementação lógica mais comum encontrada para o medidor de fase direta é mostrada na Figura 11. Nessa arquitetura, o flip-flop RS é constituído por portas *NOR*. Em seu trabalho, o autor ainda apresenta algumas variações do medidor na qual a frequência de *clock* é obtida se dividindo uma frequência de *clock* padrão do sistema por $3,6 \times 10^n$. Ainda, executa-se a operação *AND* da saída do medidor entre a saída Q do flip-flop, a frequência de *clock*, e a frequência de *clock* dividida por $3,6 \times 10^n$.

Figura 11 - Diagrama em blocos de um medidor de fase

Fonte: (RATHORE, 2003)

Para o diagrama em blocos da Figura 11, quando o sinal defasado $V_P(t)$ entra no semiciclo positivo, e o sinal de referência $V_R(t)$ está no semiciclo negativo, a saída Q do flip-flop RS comuta para valor lógico um, permitindo que os pulsos de *clock* sejam contados através da porta *AND* na saída do medidor. Quando o sinal $V_R(t)$ retorna ao semiciclo positivo e o sinal $V_P(t)$ ainda está no semiciclo positivo, a saída Q do flip-flop RS comuta para valor lógico zero, interrompendo a contagem de pulsos. Como resultado, a saída Q do flip-flop RS fica em nível lógico um apenas durante o intervalo de tempo t correspondente ao atraso da fase do sinal VP. A Figura 12 mostra os sinais $V_R(t)$ e $V_P(t)$ e a saída Q para um sinal $V_P(t)$ atrasado em 30° . Ambos os sinais $V_R(t)$ e $V_P(t)$ tem amplitude 1 volt, e frequência de 10 Hz.

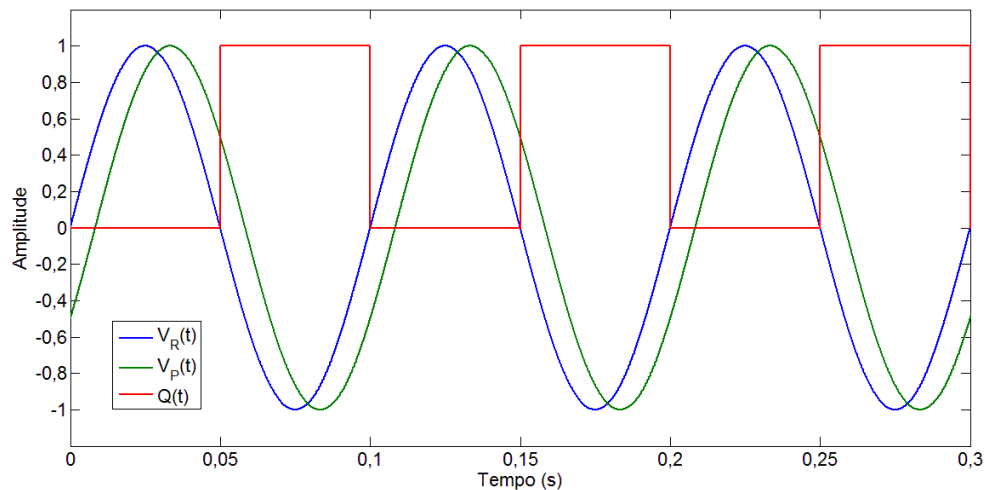
Figura 12 - Saída do flip-flop RS para um sinal $V_P(t)$ atrasado em 30°



Quando o sinal $V_P(t)$ está adiantado, a arquitetura apresentada na Figura 11 falha. Isso ocorre porque o sinal $V_R(t)$ vai para o semiciclo positivo enquanto o sinal $V_P(t)$ está no semiciclo negativo. Isso faz com que o sinal R do flip-flop seja ativado enquanto o sinal S do flip-flop está em nível lógico zero, causando a saída do flip-flop a comutar para valor lógico zero. Logo, a saída Q fica em sinal lógico um durante todo o semiciclo negativo do sinal de referência $V_R(t)$ resultando sempre em um valor de número de pulsos medidos equivalente

a 180° de defasagem. A Figura 13 mostra os sinais $V_R(t)$ e $V_P(t)$ e a saída Q para um sinal $V_P(t)$ atrasado em 30° .

Figura 13 - Saída do flip-flop RS para um sinal $V_P(t)$ adiantado em 30°



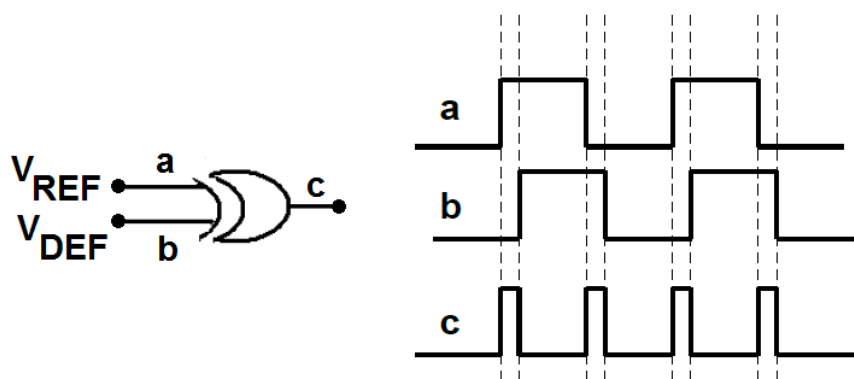
A grande desvantagem do método de contagem de pulsos é que dependendo da frequência do sinal que se deseja medir e da resolução desejada, o valor do *clock* total do sistema pode ser extremamente alto, na ordem de giga-hertz. Um ponto fraco que pode causar imprecisão é a suscetibilidade a ruído. Esse método é altamente dependente da precisão na detecção de passagem por zero do sinal medido e do sinal de referência. Qualquer ruído que cause atraso ou avanço nessa passagem irá implicar no erro de contagem do trem de pulso em um pulso de contagem para menos ou um pulso de contagem para mais, resultando em um erro direto equivalente à ordem de resolução que foi utilizada para determinar a frequência de geração dos pulsos de contagem.

2.5 TÉCNICA DA MEDIÇÃO DE FASE POR PWM

A técnica de medição de fase por PWM (modulação por largura de pulso) é uma derivação do método de detecção de fase utilizado dentro de PLLs

(*Phase-Locked Loop*). Nela a saída do detector de fase é aplicada diretamente em um filtro passa-baixa ou passa-faixa, o qual converte o trem de pulsos em uma tensão CC equivalente. Na entrada do detector de fase são utilizados um sinal de referência e um sinal a ser medido. Ambos os sinais são provenientes de um bloco detector de cruzamento por zero (*zero crossing*), o qual garante que os sinais que entram na entrada do detector de fase são ondas quadradas. Segundo Cardoso (2009), a implementação mais simples de um detector de fase é através de uma porta lógica ou-exclusivo (XOR). A Figura 14 mostra um sinal de referência a , um sinal defasado b , e a saída c de uma porta XOR.

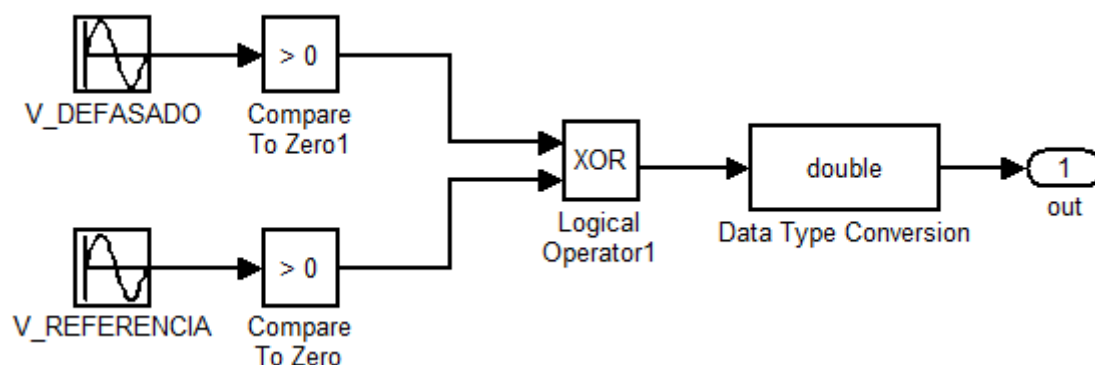
Figura 14 - Sinais de entrada e saída de um detector de fase de porta XOR



Adaptado de: (TASSARA, 2009)

De acordo com Oliveira (2005) o método de detecção de fase através de uma porta XOR só é aplicável quando o *duty-cycle* das ondas quadradas que entram no detector de fase é de 50%, e quando a diferença de fase entre os sinais varia de 0 a π radianos. A Figura 15 mostra um modelo Simulink utilizado para testar essa limitação.

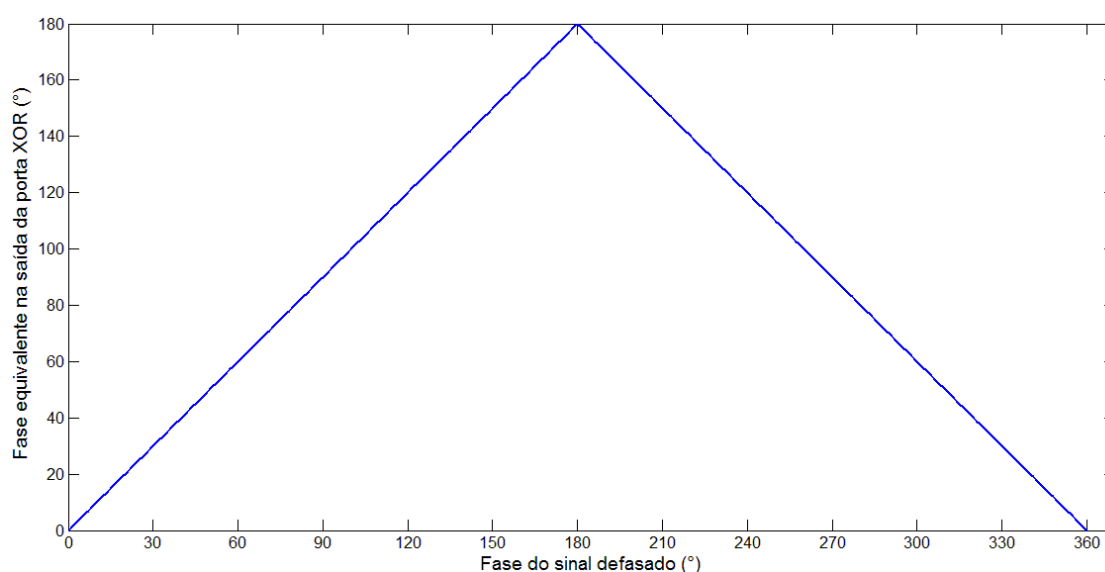
Figura 15 - Modelo Simulink de um detector de fase com porta XOR



No modelo, os comparadores com zero executam a função do detector de cruzamento por zero, e geram uma onda quadrada com *duty-cycle* de 50%, uma vez que o sinal na entrada é um sinal senoidal.

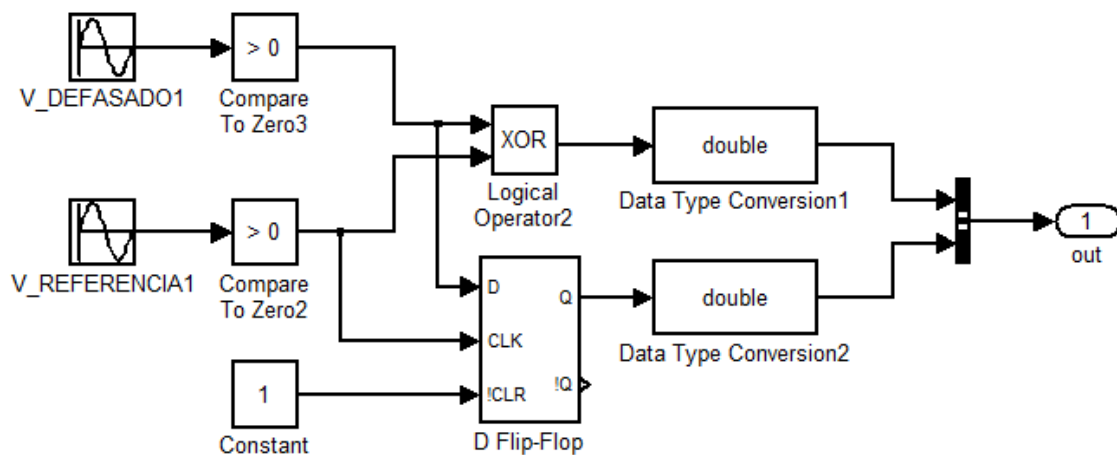
Para o teste variou-se o ângulo de defasagem do sinal V_DEFASADO de 0° até 360° com passo de 1°. O valor de fase na saída do detector de fase foi calculado com um *script* MATLAB que calcula a largura dos pulsos na saída, e as aplica na equação (72). A curva de resposta indicando a fase medida na saída do detector de fase é apresentada na Figura 16.

Figura 16 - Resposta de fase para o modelo Simulink do detector de fase com porta XOR



Observa-se que a resposta do detector de fase corresponde sempre a um valor entre 0 a π radianos. Porém, quando a diferença de fase entre os sinais é superior a π radianos, a resposta do detector é equivalente a um sinal adiantado, mostrando que a resposta do detector de fase é válida para diferenças de 0 a 2π . Uma indicação de avanço ou atraso de fase pode ser obtida com a introdução de um *flip-flop* tipo D, conforme propõe Oliveira (2005). O detector de fase com porta XOR e indicação de avanço ou atraso de fase é mostrado na Figura 17.

Figura 17 - Modelo Simulink de um detector de fase com porta XOR e indicação de avanço ou atraso de fase

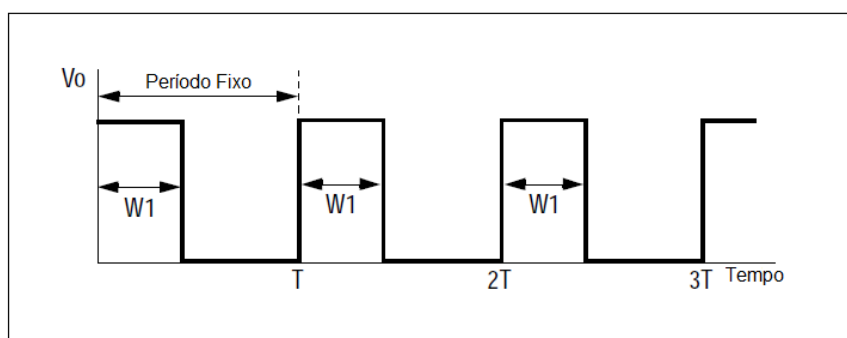


A onda quadrada referente ao sinal de referência é utilizada como *clock* para o *flip-flop* tipo D e onda quadrada referente ao sinal defasado é utilizada como o dado na entrada do *flip-flop*. Quando o sinal está atrasado, a saída do *flip-flop* é nível lógico “0” e quando o sinal está adiantado a saída do *flip-flop* é nível lógico “1”.

Em um sinal PWM típico, a frequência de base é fixa porém a largura de pulso pode ser variável ou fixa. No caso do detector de fase em uma aplicação de bioimpedância, a diferença de fase entre o sinal de referência e o sinal medido será constante, deste modo, o sinal PWM resultante será um sinal PWM com

largura de pulso fixa. A Figura 18 mostra a forma de onda de um sinal PWM típico com largura de pulso fixa.

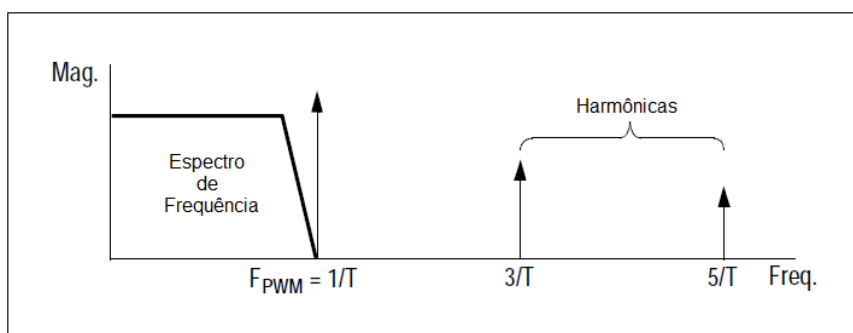
Figura 18 - Forma de onda de um sinal PWM típico com largura de pulso fixa



Fonte: (HYNIX, 2001)

De acordo com as notas de aplicação da Hynix (2001), uma análise de Fourier de um sinal PWM, mostra que existe um pico grande na frequência de base $F_n = 1/T$. Outras harmônicas fortes também existem nas frequências $F = K/T$, onde K é um inteiro. Estes picos são ruído CA indesejado e devem ser eliminados para se obter um valor CC. Normalmente isso é feito utilizando-se um filtro passa-baixa. A Figura 19 mostra a distribuição do espectro de um sinal PWM.

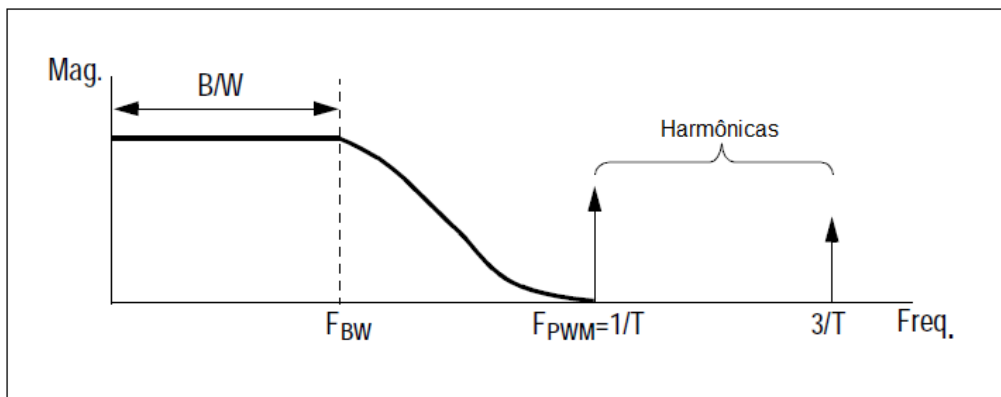
Figura 19 - Distribuição espectral de um sinal PWM.



Fonte: (HYNIX, 2001)

A faixa de banda desejada para o filtro passa baixa deve ser $F_{BW} \leq (F_{PWM} = 1/T)$. Porém, se faixa de banda for selecionada como $F_{BW} = F_{PWM}$, então o filtro passa-baixa tem que ser um filtro passa-baixa ideal, do tipo *brick-wall*. Deste modo, para propósitos práticos, o filtro passa-baixa utilizado têm uma largura de banda de corte muito menor que a largura de banda PWM, conforme mostrado na Figura 20. (HYNIX, 2001)

Figura 20 - Comparação de largura de banda entre o filtro passa-baixa e o sinal PWM



Fonte: (HYNIX, 2001)

A partir dessa consideração, são impostas as relações representadas nas equações (77) a (79). O valor de K é uma constante e deve ser escolhido de acordo com o número de decibéis que serão rejeitados da componente fundamental do ruído do sinal PWM.

$$F_{BW} \ll F_{PWM} \quad (77)$$

$$F_{PWM} = K \times F_{BW} \quad (78)$$

$$K \gg 1 \quad (79)$$

Utilizando-se o procedimento das notas de aplicação da Hynix (2001), deve-se arbitrar um valor elevado de K , e considerando a frequência do sinal na

saída do detector de fase como sendo a largura de banda F_{PWM} , deve-se determinar o valor da largura de banda do filtro F_{BW} .

O ruído deverá ser filtrado na frequência do sinal na saída do detector de fase. Sendo $H(f)$ a função de transferência complexa do filtro para a resposta em frequência, e $|H(f)|$ o módulo da função de transferência complexa do filtro. A atenuação do pico principal da frequência de saída do detector de fase é calculada de acordo com a equação (80).

$$dB = 10 \cdot \log|H(f)|^2 \quad (80)$$

O valor de tensão do ruído fundamental pode ser calculado através das equações (81) e (82), nas quais PWM_{high} é a tensão de pico do trem de pulsos que sai do detector de fase, PWM_{duty} é o *duty-cycle* do trem de pulsos em porcentagem, e PWM_{period} é o período do trem de pulsos em porcentagem (100%).

$$V_{main} = 2 \cdot PWM_{high} \cdot \sin\left(\frac{PWM_{duty}}{PWM_{period}}\right) \quad (81)$$

$$V_{noise} = 10^{\left(\frac{dB}{20}\right)} \cdot V_{main} \quad (82)$$

A tensão CC do filtro passa-baixa pode ser calculada com a equação (83).

$$V_{lowpass_DC} = PWM_{high} \cdot \left(\frac{PWM_{duty}}{PWM_{period}}\right) \quad (83)$$

A tensão de saída do filtro passa baixa é a composição da tensão CC obtida pela equação (83) e da tensão CA obtida com a equação (82). Assim, a tensão de saída do filtro é expressa pela equação (84).

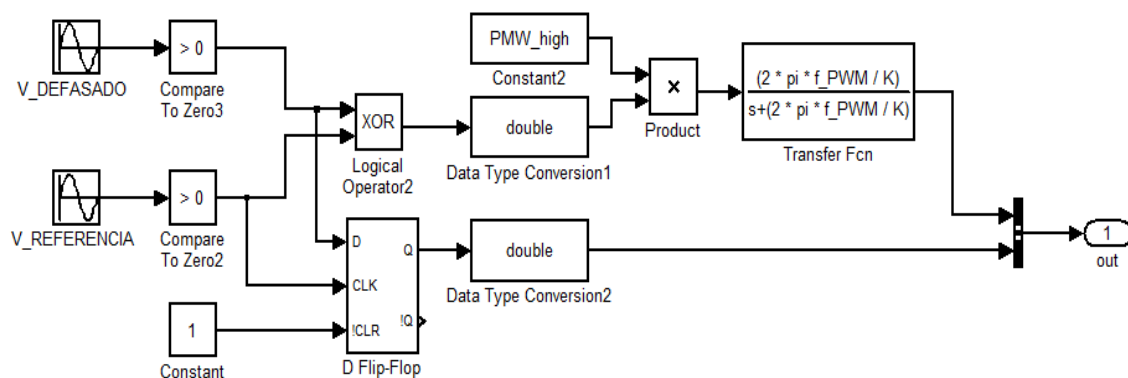
$$V_{lowpass} = V_{lowpass_DC} + V_{noise} \quad (84)$$

Para se aumentar o valor da tensão CA que é rejeitada, pode-se aumentar o valor de K , ou pode-se aumentar a ordem do filtro. Quando se projeta um filtro passa-baixa em PWM, o aumento do valor de K resulta na necessidade de um

valor de frequência fundamental maior no detector de fase. Como para a medição de bioimpedância a frequência fundamental é fixa em valores padronizados de acordo com a técnica de bioimpedância utilizada, o ajuste da rejeição CA pelo aumento do valor de K resulta em uma atenuação maior no filtro. Isso pode fazer com que a tensão CC resultante seja muito pequena e suscetível a outras fontes de ruído.

O diagrama completo de um detector de fase com porta XOR, indicação de avanço ou atraso de fase e conversão de tensão via filtro passa-baixa de primeira ordem é mostrado na Figura 21.

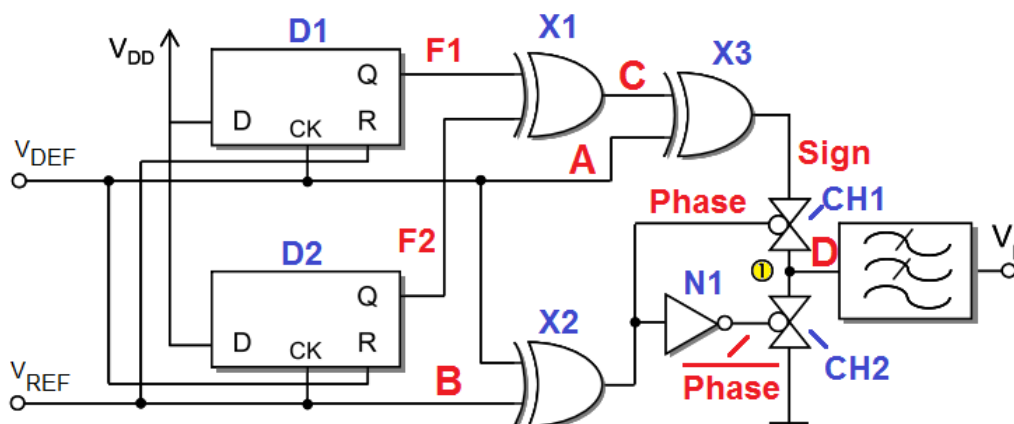
Figura 21 - Modelo Simulink de um detector de fase completo



Na implementação real de um medidor de fase por PWM é bastante comum a utilização de amplificadores operacionais na configuração de comparador não-inversor para executar a função de detecção de passagem por zero. De acordo com Lawson e Brown (1981), essa abordagem acarreta em erros de medição pois os amplificadores operacionais reais não são perfeitos, possuindo sempre pequenos erros de tensão de *offset*. Como resultado, o erro da tensão de *offset* provoca um erro de *offset* de temporização. As saídas dos amplificadores operacionais não irão mudar no instante em que o sinal na entrada do comparador passa por zero, mas sim, um pouco antes ou um pouco depois dependendo da polaridade do erro da tensão de *offset*.

Como solução para compensar o erro de temporização, que leva a perda de precisão na medição, Lawson e Brown (1981) propõe a utilização de um comparador não-inversor regenerativo e o extrator de fase digital apresentado na Figura 22. Um capacitor é associado em série ao resistor de *feedback* do comparador não-inversor regenerativo para prevenir oscilações e perturbações indesejadas enquanto que o extrator de fase digital compensa os erros de *offset* de temporização que podem ser introduzidos pelos amplificadores operacionais dos comparadores da entrada.

Figura 22 - Diagrama esquemático de um extrator de fase com correção de temporização.



Fonte: (LAWSON; BROWN, 1981)

A solução proposta na Figura 22 fornece no nó 1 uma forma de onda PWM com largura de pulso variável, sendo que o *duty-cycle* médio representa a diferença de fase real dos sinais digitais. Na arquitetura proposta, o valor lógico “1” equivale a uma tensão próxima de +VDD e o valor lógico “0”, equivale a uma tensão -VSS diferente de zero. O ponto terra (*ground*) equivale a tensão 0V.

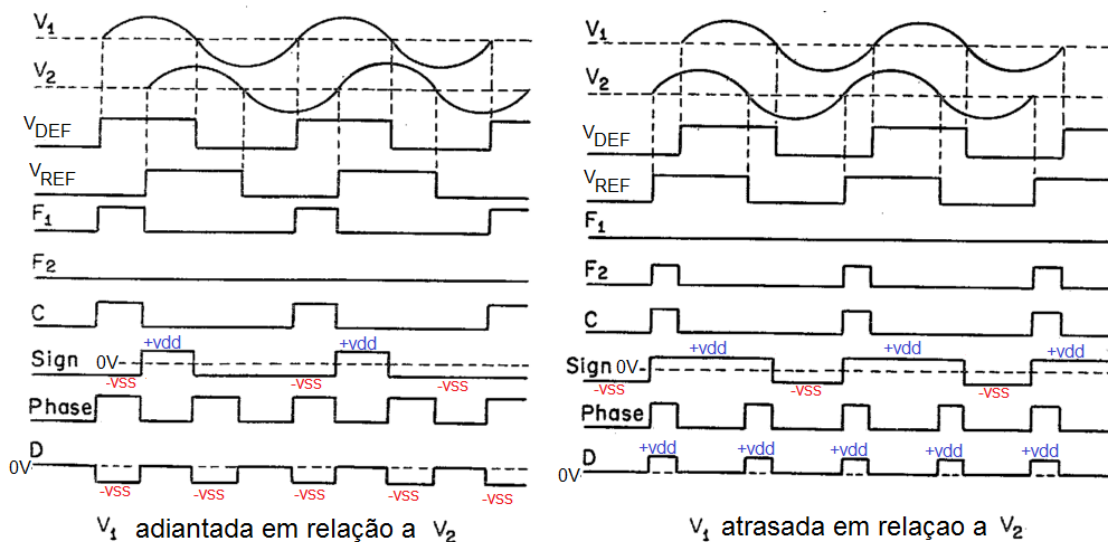
Da mesma forma como na arquitetura tradicional apresentada na Figura 15 e na Figura 16, o nível lógico “1” do sinal Phase representa o período de tempo equivalente a diferença de fase entre os sinais V_{DEF} e V_{REF} , e é obtido diretamente por uma porta XOR (porta X2). O sinal Sign indica avanço de fase (sinal lógico “0”) ou atraso de fase (sinal lógico “1”). A obtenção do sinal Sign, é

realizada através da lógica dos flip-flops D1 e D2, e das portas XOR X1 e X3. O valor lógico do sinal Sign permite gerar uma tensão positiva no nó 1 quando a existe atraso de fase e uma tensão negativa quando existe avanço de fase.

Devido a ligação dos sinais V_{REF} e V_{DEF} aos *flip-flops* D1 e D2, o sinal V_{REF} quando está em nível lógico “1” reseta o *flip-flop* D1, e o sinal V_{DEF} quando está em nível lógico “1” reseta o *flip-flop* D2. Deste modo, o *clock* só comuta os *flip-flops* para um valor diferente de nível lógico “0”, apenas quando um dos sinais está em nível lógico “0” e o outro está em nível lógico “1”. Se ambos V_{REF} e V_{DEF} comutam para nível lógico zero, os *flip-flops* D1 e D2 não tem a saída alterada. Logo, o *flip-flop* D1 só assume valor lógico “1” durante o intervalo de tempo no qual o sinal V_{DEF} está adiantado em relação ao sinal V_{REF} . De forma semelhante, o *flip-flop* D2 só assume valor lógico “1” quando o sinal V_{DEF} está atrasado em relação ao sinal V_{REF} . A Figura 23 apresenta as saídas dos *flips-flops* D1 e D2, sinais F1 e F2 respectivamente, para o avanço e o atraso de V_{DEF} . O sinal V_{DEF} é resultado do comparador não-inversor para a onda senoidal defasada V_1 e o sinal V_{REF} é resultado do comparador não-inversor para a onda senoidal de referência V_2 .

A porta XOR X1 garante que o sinal de referência C, estará em nível lógico “1” apenas quando ou o sinal F1 estiver em nível lógico “1” ou apenas quando o sinal F2 estiver em nível lógico “1”. Logo, o sinal C estará em nível lógico “1” apenas durante o intervalo de tempo correspondente a defasagem entre os sinais V_{DEF} e V_{REF} conforme mostrado na Figura 23. O sinal de referência C é aplicado a porta XOR X3 juntamente com o sinal V_{DEF} para a produção do sinal Sign. O sinal Sign só tem efeito quando o sinal Phase tem nível lógico “1”, permitindo que a chave analógica CH1 seja ativada e a tensão correspondente ao valor lógico do sinal Sign seja conectada ao ponto D. Assim, durante o intervalo de tempo correspondente a defasagem entre os sinais V_{REF} e V_{DEF} , quando o sinal Phase tem nível “lógico 1”, se o sinal V_{DEF} está atrasado, o sinal Sign tem nível lógico “1” e a tensão no ponto D será +VDD. De modo similar, quando o sinal V_{DEF} está adiantado, o sinal Sign tem nível lógico “0” e a tensão no ponto D será -VSS. Quando o sinal Phase tem valor lógico “0”, o inversor N1 comuta a chave analógica CH2 fazendo com que o ponto D seja conectado ao sinal terra, resultando em uma tensão de 0V.

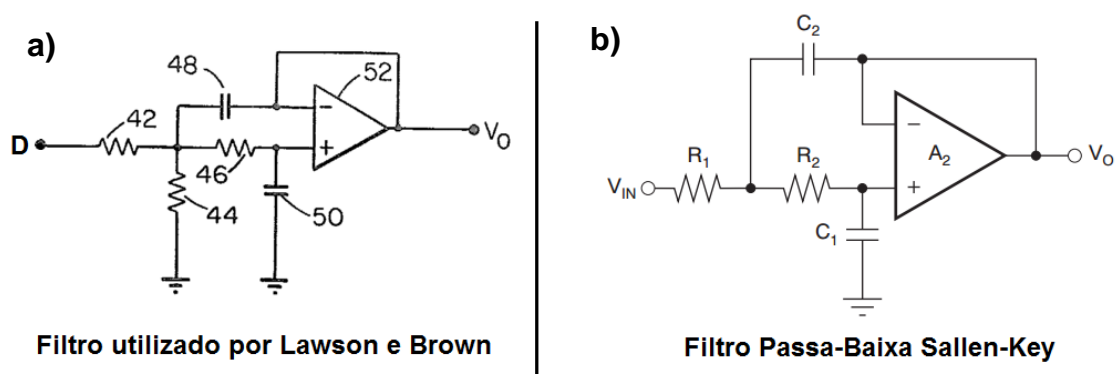
Figura 23 - Formas de onda do extrator de fase, para avanço e atraso de do sinal senoidal defasado



Fonte: (LAWSON; BROWN, 1981)

O filtro passa baixa utilizado por Lawson e Brown (1981), mostrado na Figura 24a é um filtro passa-baixa de dois polos em uma configuração de componentes que se assemelha ao filtro passa-baixa de topologia Sallen-Key.

Figura 24 - Comparação de topologia de filtros de segunda ordem

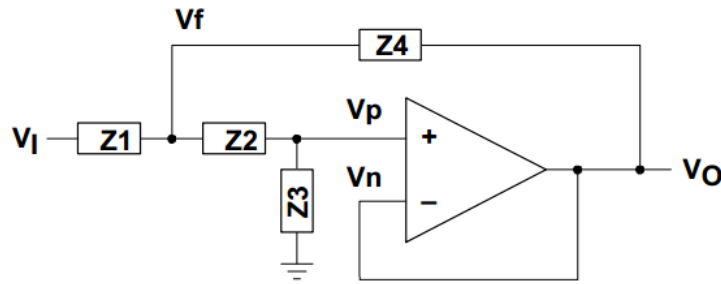


Fonte: (LAWSON; BROWN, 1981)

Fonte: (TI, 2011)

A topologia do filtro Sallen-Key mostrada na Figura 24b é uma topologia de ganho unitário. Seguindo o desenvolvimento mostrado por Karki (2002), a forma generalizada dessa topologia pode ser expressa em termos de impedâncias genéricas conforme é ilustrado na Figura 25.

Figura 25 - Diagrama generalizada para impedâncias.



Fonte: (KARKI, 2002)

Para se encontrar a solução da forma generalizada do circuito deve-se encontrar as relações entre as tensões V_i , V_o , V_p e V_n . A aplicação das leis de Kirchhoff para o nó V_f resulta na equação (85) e a aplicação das leis de Kirchhoff para o nó V_p resulta na equação (86).

$$V_f \cdot \left(\frac{1}{Z_1} + \frac{1}{Z_2} + \frac{1}{Z_4} \right) = V_i \cdot \left(\frac{1}{Z_1} \right) + V_p \cdot \left(\frac{1}{Z_2} \right) + V_o \cdot \left(\frac{1}{Z_4} \right) \quad (85)$$

$$V_p \cdot \left(\frac{1}{Z_2} + \frac{1}{Z_3} \right) = V_f \cdot \left(\frac{1}{Z_2} \right) \Rightarrow V_f = \left(1 + \frac{Z_2}{Z_3} \right) \quad (86)$$

Conforme mostrado por Karki (2002), se as equações (86) e (85) forem resolvidas para V_p , e a equação resultante for correlacionada com blocos de ganhos, pode-se obter a função de transferência do filtro no domínio do tempo em função de um ganho K , conforme expressa a equação (87).

$$\frac{V_o}{V_i} = \frac{K}{\frac{Z_1 \cdot Z_2}{Z_3 \cdot Z_4} + \frac{Z_1}{Z_3} + \frac{Z_2}{Z_3} + \frac{Z_1 \cdot (1 - K)}{Z_4} + 1} \quad (87)$$

Se os valores de impedância na equação (87) forem substituídos pelos valores correspondentes em função dos componentes utilizados na topologia apresentada na Figura 24b, a equação de transferência do filtro Sallen-Key no domínio da frequência será a equação (88).

$$\frac{V_o}{V_i} = \frac{K}{s^2 \cdot (R_1 \cdot R_2 \cdot C_1 \cdot C_2) + s \cdot (R_1 \cdot C_1 + R_2 \cdot C_1 + R_2 \cdot C_2 \cdot (1 - K)) + 1} \quad (88)$$

A partir da função de transferência no domínio da frequência de um filtro passa-baixa genérico de segunda ordem, mostrada na equação (89), pode-se obter a correlação entre o valor dos componentes da topologia com os parâmetros de projeto de um filtro.

$$H_{LP}(s) = \frac{K}{s^2 \cdot \left(\frac{1}{2 \cdot \pi} \cdot \frac{1}{f_c}\right)^2 + s \cdot \left(\frac{1}{2 \cdot \pi} \cdot \frac{1}{Q \cdot f_c}\right) + 1} \quad (89)$$

Na equação (89) K é o ganho, Q é o fator de qualidade e f_c é a frequência do ponto de transição entre a banda passante e a banda rejeitada, não sendo necessariamente o ponto de -3 dB de atenuação que normalmente é utilizado para caracterizar a frequência de corte. Igualando-se os termos da equação (88) com os termos da equação (89), a frequência de transição e o fator de qualidade são expressos pelas equações (90) e (91) respectivamente.

$$f_c = \frac{1}{2 \cdot \pi \cdot \sqrt{R_1 \cdot R_2 \cdot C_1 \cdot C_2}} \quad (90)$$

$$Q = \frac{\sqrt{R_1 \cdot R_2 \cdot C_1 \cdot C_2}}{R_1 \cdot C_1 + R_2 \cdot C_1 + R_1 \cdot C_2 \cdot (1 - K)} \quad (91)$$

Para o projeto do filtro são aplicadas razões de proporcionalidade entre os componentes para se simplificar o procedimento de cálculo. Obtém-se então as relações simplificadas para as equações (90) e (91), equações (96) e (97) respectivamente, com base nas constantes de proporcionalidade mostradas nas equações (92) a (95).

$$R_1 = m \cdot R \quad (92)$$

$$R_2 = R \quad (93)$$

$$C_1 = C \quad (94)$$

$$C_2 = n \cdot C \quad (95)$$

$$f_c = \frac{1}{2 \cdot \pi \cdot R \cdot C \sqrt{m \cdot n}} \quad (96)$$

$$Q = \frac{\sqrt{m \cdot n}}{m + 1 + m \cdot n \cdot (1 - K)} \quad (97)$$

Devido a interação entre f_c e Q causada pelo fato de ambas as equações (94) e (95) serem dependentes dos coeficientes m e n , o projeto deve ser feito escolhendo-se um valor de ganho K e um fator de qualidade Q para a determinação das constantes m e n . Escolhe-se então um valor para o capacitor C , e ajusta-se o valor de R para se obter a frequência f_c . O valor máximo do ganho K que pode ser utilizado é limitado pela equação (96).

$$K < 1 + \frac{m + 1}{m \cdot n} \quad (98)$$

Quando o valor do ganho é igual ao limite imposto pela equação (98), ocorre uma divisão por zero na equação (97) e o valor do fator de qualidade é infinito. Quando o ganho ultrapassa o limite imposto pela equação (98), ocorre divisão por um número negativo na equação (97) e o valor do fator de qualidade se torna negativo, fazendo com que o polo se desloque no plano s para uma posição na qual o circuito começa a oscilar.

Se ambos os coeficientes de proporcionalidade forem escolhidos com valor igual a um, os resistores serão iguais e os capacitores serão iguais. Como resultado, as equações para f_c e o fator de qualidade Q são reduzidas as equações (99) e (100), as quais são independentes uma da outra.

$$f_c = \frac{1}{2 \cdot \pi \cdot R \cdot C} \quad (99)$$

$$Q = \frac{1}{3 - K} \quad (100)$$

Para essa aproximação, o fator de qualidade Q é função apenas do ganho do filtro, e escolhendo-se um valor para o capacitor C , pode-se calcular facilmente o valor do resistor R para se obter o valor de f_c . Esse procedimento de parametrização do filtro de topologia Sallen-Key pode ser aplicado para qualquer topologia de filtro que se deseje utilizar junto com a técnica de determinação de fase por PWM.

2.6 TÉCNICA DA MEDIÇÃO POR DFT

A técnica de medição pela transformada discreta de Fourier - DFT (*Fourier Discret Transform*) permite a determinação tanto da magnitude quanto da fase do sinal medido. No trabalho de Amaral et al. (2011) é apresentado um medidor de bioimpedância a partir de DFT implementado em DSP, o qual permite a medição de uma banda de frequência de 50 kHz até 500 kHz e componentes resistivas de 1 Ω até 50 k Ω . O medidor utiliza a técnica de *undersampling* para permitir que sinais de altas frequências sejam analisados, uma vez que a frequência máxima de amostragem do ADC utilizado era de 12,56 kHz.

A técnica de *undersampling* é baseada no princípio de que mesmo quando o sinal é amostrado em uma frequência inferior à frequência do critério de Nyquist, as amostras obtidas ainda contêm a mesma informação de fase e amplitude da forma de onda original. Segundo Amaral et al. (2011) o uso de uma frequência de amostragem única durante o processo de *undersampling* é mais suscetível ao erro induzido por harmônicas. Assim, para aumentar a precisão da medida é utilizado um período de amostragem variável.

A transformada de Fourier representa uma função como sendo composta por uma soma de exponenciais complexas representativas de magnitudes, frequências e fases permitindo a análise no domínio da frequência. (Oliveira, 2005). A transformada discreta de Fourier pode ser utilizada para sinais periódicos ou discretos. Na prática, a DFT é implementada com o uso do

algoritmo denominado transformada rápida de Fourier – FFT (*Fast Fourier Transform*), que é um algoritmo computacionalmente eficiente por reduzir a complexidade de cálculo da DFT da ordem de N^2 para $N \cdot \log_2 N$. Exemplificando, para calcular a DFT de 1000 amostras, o cálculo pelo método teórico da DFT necessitaria de 10^6 operações enquanto que a FFT necessitaria de 9966 operações.

O desenvolvimento e implementação de um algoritmo FFT em FPGA é mostrado por Bonilla et al. (2012). O algoritmo desenvolvido é do tipo Radix-4 e utiliza menos *slices* do FGPA quando comparado ao algoritmo do tipo Radix-2 (o algoritmo mais conhecido), e quando comparado com *ip-core* de FFT versão 7.1 da Xlink. O algoritmo Radix-4 foi implementado para processar 1024 amostras complexas, sendo que cada amostra tem 64 *bits* (32 *bits* para a parte real e 32 *bits* para a parte imaginária). O formato do número de 32-bits é no padrão de precisão simples IEEE-754 (1 *bit* de sinal, 8 *bits* para o expoente de 8 *bits* e 23 *bits* para a parte fracionária). A quantidade de recursos utilizados pela implementação do algoritmo Radix-4 em um dispositivo Spartan-3 é mostrada na Tabela 3.

Tabela 3 - Utilização de recursos do algoritmo Radix-4 em um FPGA Spartan-3

Utilização Lógica	Usado	Disponível	Utilização (%)
Número de Slice Registers	419	4800	8
Número de Slice LUTs	1202	2400	50
Número de fully used LUT-FF pairs	147	1474	9
Número de bonded IOBs	7	102	4
Número de Block RAM/FIFO	10	12	8
Número de BUFG/BUFGCTRLs	1	16	6

Fonte: (BONILLA et al., 2012)

Dado um sinal $x(t)$ contínuo no domínio no tempo, a sequência discreta no domínio do tempo que descreve os valores amostrados pode ser denotada

como $x[n]$ e o número total de amostras da sequência de entrada pode ser denotada como N . Se o sinal $x(t)$ é um sinal periódico, o sinal $x[n]$ também o será. No domínio da frequência, existirá um número finito de harmônicos que formam uma combinação linear de exponenciais complexas que sintetizam o sinal discreto $x[n]$. Desse modo, a sequência periódica $X[k]$ no domínio da frequência será os coeficientes da série discreta de Fourier que sintetiza o sinal discreto $x[n]$. A equação (101), denominada equação de análise, é a equação que transforma o sinal discreto no domínio do tempo para o sinal discreto no domínio da frequência. A equação (102), denominada equação de síntese, é a equação que transforma o sinal discreto no domínio da frequência para o sinal discreto no domínio do tempo. A equação (103) apresenta a relação da exponencial complexa. A equação (104) apresenta a sequência genérica de harmônicos que sintetizam o sinal discreto $x[n]$.

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot (W_N)^{k \cdot n} \quad (101)$$

$$x[n] = \frac{1}{N} \cdot \sum_{k=0}^{N-1} X[k] \cdot (W_N)^{-k \cdot n} \quad (102)$$

$$W_N = e^{-j \cdot \left(\frac{2 \cdot \pi}{N}\right)} = \quad (103)$$

$$\text{frequências} \left\{ \left(\frac{2 \cdot \pi}{N}\right) \cdot k, \text{ para } k = 0, 1, 2, \dots, N-1 \right\} \quad (104)$$

Através da relação de Euler, a equação (101) pode ser escrita em função de seno e cosseno ao invés de exponencial, o que é vantajoso quando se implementa o cálculo através de tabelas de seno e cosseno (uma tabela de seno e cosseno é uma tabela de coeficientes armazenados na memória). A equação (105) é a equação de análise reescrita em função de seno e cosseno.

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot \left[\cos\left(\frac{2 \cdot \pi \cdot k \cdot n}{N}\right) - j \cdot \sin\left(\frac{2 \cdot \pi \cdot k \cdot n}{N}\right) \right] \quad (105)$$

A série periódica no domínio da frequência $X[k]$ será composta por coeficientes complexos. Tomando-se os coeficientes complexos na forma polar,

o coeficiente com a maior magnitude estará na frequência fundamental do sinal amostrado. O valor da maior magnitude será proporcional a amplitude máxima do sinal contínuo $x(t)$ no domínio do tempo. De forma similar, o ângulo do coeficiente da DFT na frequência fundamental será proporcional a fase do sinal contínuo $x(t)$ no domínio do tempo.

Utilizando-se do efeito da proporcionalidade de magnitude e de fase entre o coeficiente da DFT na frequência fundamental e o sinal $x(t)$ no domínio do tempo, os coeficientes da DFT na frequência fundamental de dois sinais amostrados permitem a determinação da razão de amplitude e a diferença de fase entre os dois sinais. Considerando-se que um dos sinais é um sinal de referência V_{REF} , cujo coeficiente da DFT na frequência fundamental é $M_{REF}\angle\varphi_{REF}$, e o outro sinal é um sinal defasado V_{DEF} cujo coeficiente da DFT na frequência fundamental é $M_{DEF}\angle\varphi_{DEF}$, a razão entre a amplitude dos dois sinais pode ser obtida com a equação (106), e a diferença de fase pode ser obtida com a equação (107).

$$\frac{V_{DEF}}{V_{REF}} = \frac{M_{DEF}}{M_{REF}} \quad (106)$$

$$\phi_{DEF} - \phi_{REF} = \varphi_{DEF} - \varphi_{REF} \quad (107)$$

A técnica de medição de fase por DFT da forma como é mostrada na literatura, calcula a transformada de Fourier para um número grande de frequências dentro de uma banda espectral bastante larga. Isso torna esse método mais adequado para o uso em aplicações que trabalham com a aplicação de múltiplas frequências dentro do mesmo intervalo temporal de teste. Quando apenas uma única frequência está sendo testada, pode-se tentar calcular a DFT para apenas a harmônica fundamental, porém, a necessidade de cálculo com números complexos, introduz uma complexidade matemática que não é a mais indicada para aplicações mais simples que não utilização análise espectral para garantir a segurança dos resultados.

2.7 TÉCNICA DA MEDIÇÃO DE FASE POR AJUSTE DE SENOIDE

De acordo com Zhang et al. (1997), as rotinas e algoritmos de ajuste de senóide são usadas extensivamente durante a caracterização de conversores ADC e osciloscópios digitais (levantamento de resposta à aplicação de um sinal de referência). O algoritmo de ajuste de senóide mais difundido é o algoritmo de quatro parâmetros que é utilizado para obter frequência, amplitude e fase de uma onda senoidal. Existe uma extensão do algoritmo de quatro parâmetros, denominado algoritmo de sete parâmetros que serve para aplicações de duplo canal nas quais se deve estimar os parâmetros de dois sinais de mesma frequência. (RAMOS et al., 2010)

Uma variação do algoritmo de quatro parâmetros com desempenho superior é proposta por Zhang et al. (1997). O aumento de desempenho é obtido substituindo-se o método dos mínimos quadrados pelo método dos mínimos quadrados totais, obtendo-se um erro residual menor no ajuste da curva.

Supondo-se um sinal senoidal contínuo no domínio do tempo que se comporta de acordo com a equação (108), se ele for amostrado em um intervalo de amostragem T_s , então os valores amostrados podem ser representados de acordo com a equação (109). Onde $e(n)$ é um termo aditivo devido ao ruído do processo de amostragem e $\Omega = \omega \cdot T_s$ é a frequência digital correspondente a ω .

$$f(t) = D + A \cdot \text{sen}(\omega \cdot t + \theta) \quad (108)$$

$$f(n) = D + A \cdot \text{sen}(n \cdot \Omega + \theta) + e(n), \quad \text{para } n = 0, 1, 2, \dots, N \quad (109)$$

Devido a formulação do problema através da técnica do método dos mínimos quadrados totais, o conjunto de valores amostrados $f(n)$ e a equação (109), resultam na equação matricial (110), a qual representa um sistema de equações sobredeterminado. A solução desse tipo de sistema através de técnica de mínimos-quadrados é feita através das equações (111) e (112), as quais utilizam a matriz de coeficientes transposta, para se obter um sistema 3x3. O sistema 3x3 é resolvido segundo a equação (113).

$$\underbrace{\begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(N) \end{bmatrix}}_{\vec{b}} \approx \underbrace{\begin{bmatrix} 0 & 1 & 1 \\ \text{sen } \Omega & \cos \Omega & 1 \\ \vdots & \vdots & \vdots \\ \text{sen}(N \cdot \Omega) & \cos(N \cdot \Omega) & 1 \end{bmatrix}}_{\mathbf{mA}} \cdot \underbrace{\begin{bmatrix} A \cdot \cos \theta \\ A \cdot \text{sen } \theta \\ D \\ \vec{c} \end{bmatrix}}_{\vec{c}} \quad (110)$$

$$\mathbf{KmA} = \mathbf{mA}^T \cdot \mathbf{mA} \quad (111)$$

$$\overrightarrow{Kb} = \mathbf{mA}^T \cdot \vec{b} \quad (112)$$

$$\vec{c} = \mathbf{KmA}^{-1} \cdot \overrightarrow{Kb} \quad (113)$$

Uma vez que o vetor \vec{c} foi determinado, a amplitude A do sinal senoidal, a fase θ e a componente CC, (D), podem ser obtidas com a equação (114), a equação (115) e a equação (116) respectivamente.

$$A = \sqrt{c_1^2 + c_2^2} \quad (114)$$

$$\theta = \tan^{-1} \left(\frac{c_2}{c_1} \right) \quad (115)$$

$$D = c_3 \quad (116)$$

Quando se trabalha especificamente com sinais senoidais, o método do ajuste de senóide para se determinar a amplitude e a fase do sinal medido se mostra como uma opção bastante interessante, pois, apresenta uma maior imunidade ao ruído em relação aos métodos de medição direta, Ainda, segundo os resultados obtidos por Zhang et al. (1997), esse método pode ser bastante rápido de acordo com o número total de amostras e consegue bons resultados para uma resolução digital maior que três em *bits*.

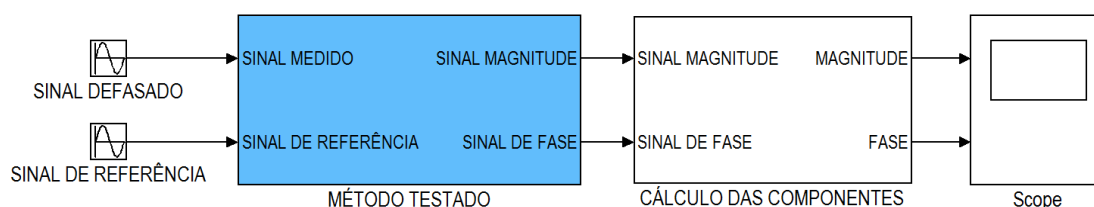
3. SIMULAÇÃO DOS MÉTODOS DE MEDIÇÃO EM SIMULINK

Este capítulo apresenta a simulação em Simulink dos métodos de medição de fase e de amplitude descritos anteriormente. São apresentadas considerações para o uso do Simulink, os modelos utilizados nas simulações e os resultados obtidos com cada método.

3.1 ARRANJO EXPERIMENTAL NO SIMULINK

A Figura 26 mostra conceitualmente o arranjo experimental utilizado no Simulink para se executar a simulação e verificação teórica dos métodos de medição de fase e de magnitude estudados e apresentados no capítulo 2.

Figura 26 - Arranjo experimental no Simulink para simulação e verificação dos métodos de medição de magnitude e fase.



No arranjo experimental utilizado um bloco gerador de sinal é utilizado como fonte de um sinal defasado de teste a ser medido, enquanto um segundo bloco gerador de sinal é utilizado como fonte de um sinal de referência com amplitude máxima unitária e fase 0° . Os sinais gerados são inseridos nos blocos do modelo Simulink que são responsáveis por efetuar o processamento segundo o método de medição que está sendo testado.

O método testado produz um sinal que representa a magnitude do sinal defasado e um sinal que representa a fase do sinal defasado. Esses sinais não são diretamente os valores medidos de fase e de magnitude do sinal defasado,

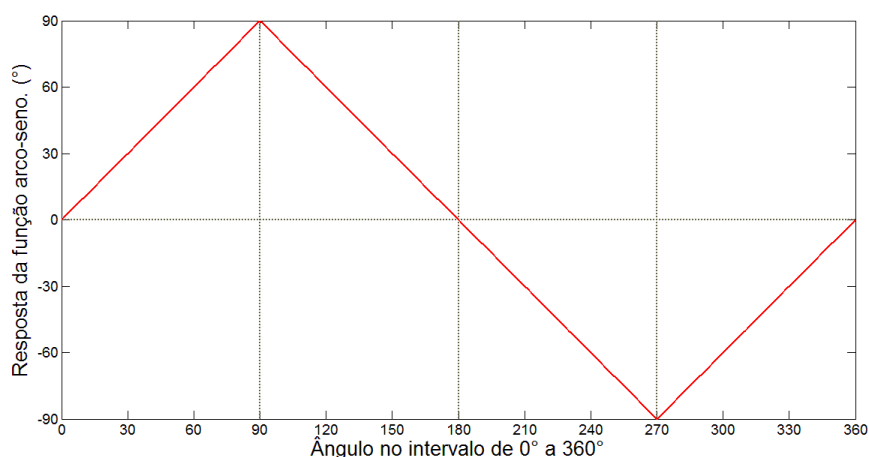
eles são na verdade sinais que de alguma forma (a qual depende do método sendo testado) representam proporcionalmente a magnitude e a fase do sinal medido. Dessa forma, o sinal de magnitude e o sinal de fase são aplicados a um bloco que tem a finalidade de calcular o valor real da magnitude e da fase medidas para o sinal defasado. Os ruídos de *hardware* não são simulados.

3.2 CONSIDERAÇÕES GERAIS PARA USO DO SIMULINK

Para a utilização do Simulink durante as simulações de medição de fase é necessário levar em conta algumas considerações relacionadas ao uso das funções de arco-trigonométrico nativas do Simulink, bem como aspectos relacionados ao tipo de *solver* e passo de tempo máximo utilizado.

As funções nativas do Simulink: arco-seno (*asin*, *asind*), e arco-tangente (*atan*, *atand*), tem resposta apenas em dois quadrantes limitando o ângulo retornado entre o intervalo $[-\pi/2, \pi/2]$. As funções arco-cosseno (*acos*, *acosd*) tem resposta em dois quadrantes retornando valores no intervalo $[0, \pi]$. A única função de arco-trigonométrico que possui resposta nos quatro quadrantes é o arco-tangente de quatro quadrantes (*atan2*, *atand2*), o qual retorna valores no intervalo $[-\pi, \pi]$. A Figura 27 mostra o ângulo retornado pela função arco-seno para todo intervalo de 0 a 360°.

Figura 27 - Resposta da função arco-seno para o intervalo $[0, 2\pi]$



A Figura 28 mostra o ângulo retornado pela função arco-tangente para todo intervalo de 0 a 360°, e a Figura 29 mostra o ângulo retornado pela função arco-cosseno. A Figura 30 mostra o ângulo retornado pela função arco-tangente de quadrantes.

Figura 28 - Resposta da função arco-tangente para o intervalo $[0, 2\pi]$

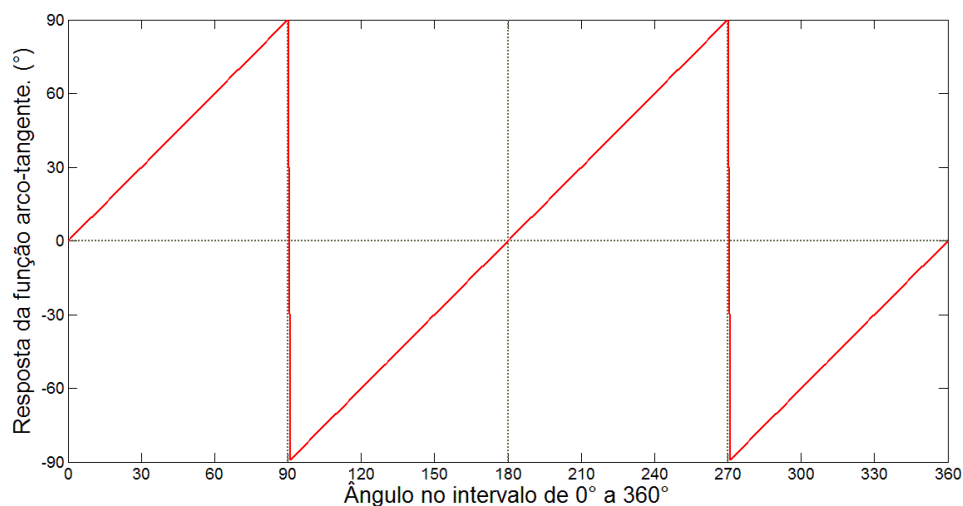


Figura 29 - Resposta da função arco-cosseno para o intervalo $[0, 2\pi]$

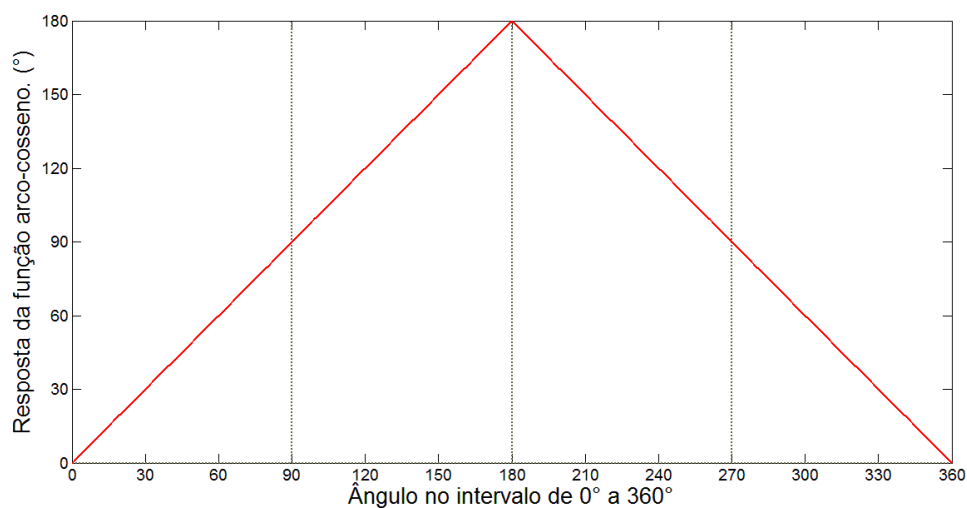
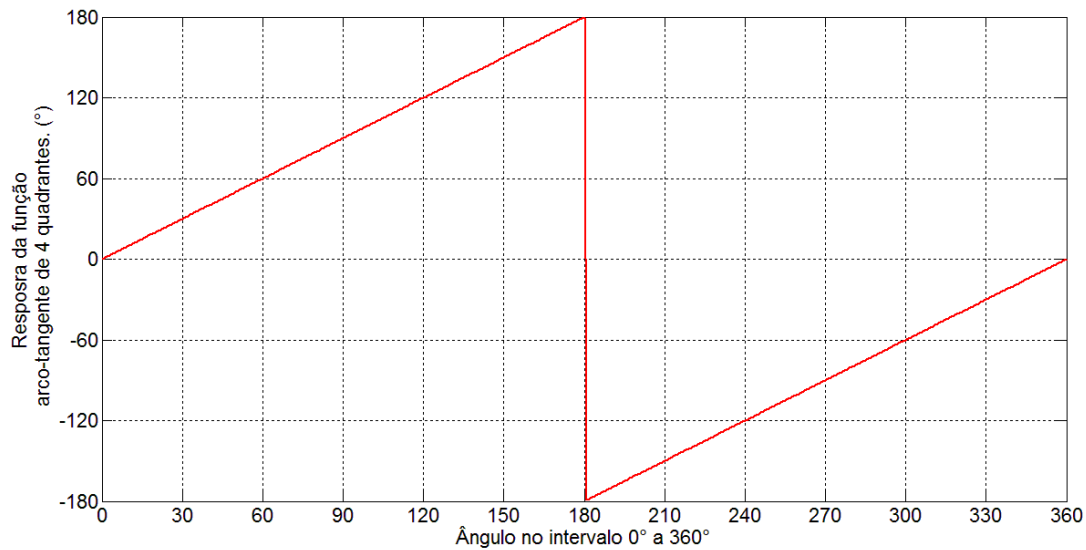


Figura 30 - Resposta da função arco-tangente de quatro quadrantes para o intervalo $[0, 2\pi]$



O valor padrão do tamanho do passo de tempo máximo utilizado por qualquer *solver* contínuo no tempo do Simulink, quando este é configurado para automático, é de 50 avos do tempo total de simulação. Para *solvers* discretos, esse valor é o menor valor de tempo de amostragem especificado no modelo. Quando se tem no modelo sinais periódicos cujos períodos são conhecidos, recomenda-se configurar o valor como sendo uma fração do menor período, como por exemplo, um quarto.

Para se determinar um passo adequado, o tamanho do passo de tempo máximo foi definido em relação a um fator de discretização da maior frequência utilizada no modelo testado. O fator de discretização utilizado é a constante β mostrada na equação (117).

$$passo_{max} = \frac{1}{\beta} \cdot \frac{1}{frequencia} \quad (117)$$

Para se obter uma discretização mínima aceitável no Simulink, o critério de Nyquist ($\beta = 2$) por si só é insuficiente, desse modo, multiplicasse o fator de discretização β por 2, para se respeitar o critério de Nyquist, e adiciona-se uma potência de 10, conforme mostrado na equação (118), para definir-se um passo menor.

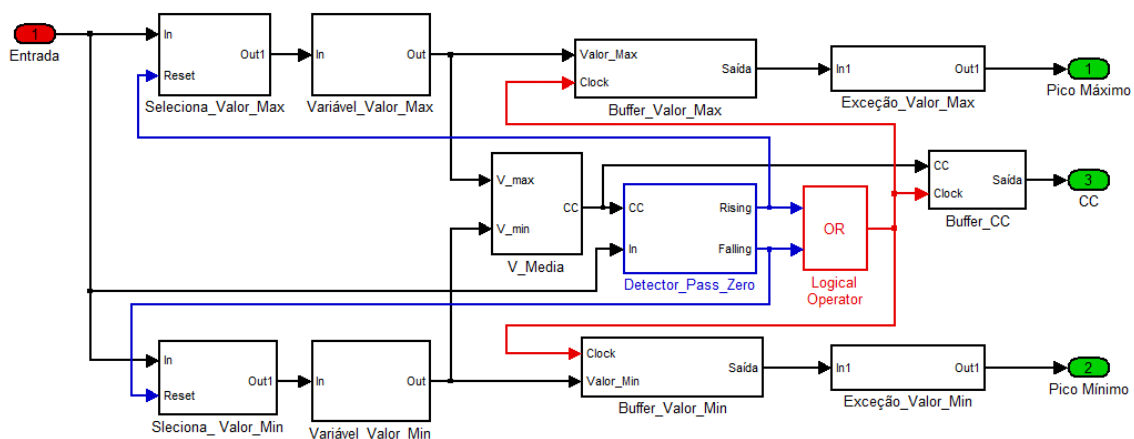
$$passo_{max} = \frac{1}{2 \cdot \beta \cdot 10^N} \cdot \frac{1}{frequencia} \quad (118)$$

O *solver* do tipo com passo de tempo variável altera o valor do passo de tempo para atingir a tolerância especificada, também diminuindo o tempo de simulação. O *solver* de passo de tempo fixo é resolvido periodicamente no passo de tempo informado ao *solver* não melhorando o desempenho.

3.3 DETECÇÃO DE MAGNITUDE DE PICO E DE VALE

A medição de bioimpedância necessita da medição simultânea da diferença de fase e da amplitude do sinal defasado. Dentre os métodos de medição apresentados anteriormente, a técnica do ajuste de elipse, a técnica do ajuste da senóide, a técnica da demodulação de quadratura e a técnica da DFT, medem simultaneamente a diferença de fase e a razão de amplitudes. Já a técnica da figura de Lissajous, a técnica da medição digital direta e a técnica de PWM necessitam de alguma técnica complementar para a detecção da magnitude do pico positivo e do pico negativo do sinal defasado. Sendo assim, desenvolveu-se o detector de magnitude de pico e componente CC mostrado na Figura 31.

Figura 31 - Modelo do detector de magnitude de pico e componente CC



Na inicialização do modelo, o valor da amplitude máxima é inicializado com o menor valor numericamente possível dentro do bloco “Variável_Valor_Max”, e o valor da amplitude mínima é inicializado com o maior valor numericamente possível dentro do bloco “Variável_Valor_Min”. Tanto o maior valor numericamente possível quanto o menor valor numericamente possível, possuem um valor em módulo fora da faixa de amplitude de operação do sinal de entrada. Como resultado, os maiores valores numericamente possíveis podem ser utilizados para reinicializar os valores de pico sem causar prejuízo a medição do valor real.

O sinal de entrada é simultaneamente comparado com o valor da amplitude máxima e com o valor da amplitude mínima, dentro dos blocos “Variável_Valor_max” e “Variável_Valor_min”. Quando a amplitude do sinal de entrada é maior que o valor da amplitude máxima, ou é igual ao menor valor numericamente possível, o valor da amplitude máxima é atualizado. Quando a amplitude do sinal de entrada é menor que o valor da amplitude mínima, ou igual ao maior valor numericamente possível, o valor da amplitude mínima é atualizado.

O bloco “V_Media” calcula a média entre o maior valor de amplitude e o menor valor de amplitude, que é a componente CC do sinal de entrada. A componente CC calculada é utilizada pelo bloco “Detector_Pass_Zero” para determinar se o sinal de entrada cruzou o nível CC. Quando o sinal de entrada cruza o patamar da componente CC de um valor negativo para um valor positivo, o sinal “Rising” vai a nível lógico “1” e quando o sinal de entrada cruza o patamar da componente CC de um valor positivo para um valor negativo, o sinal “Failling” vai a nível lógico “1”. Ambos os sinais “Failling” e “Rising” quando mudam para nível lógico “1”, produzem um sinal de *clock* que sinaliza aos *buffers* de saída para atualizarem os valores da magnitude de pico e da componente CC.

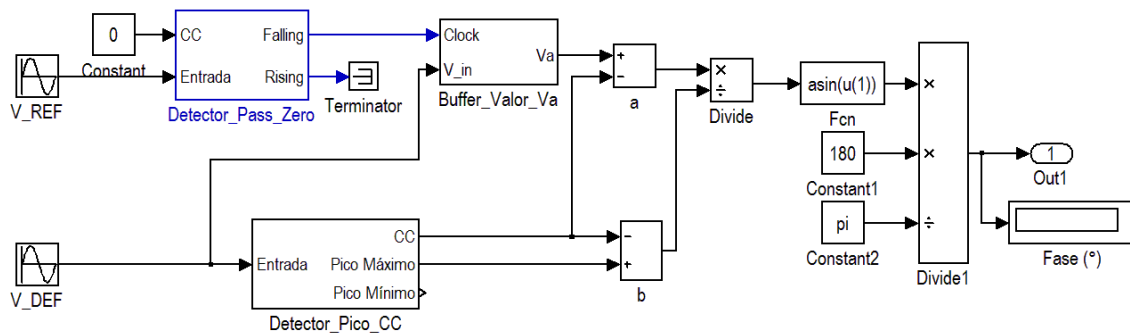
Para evitar que o detector fique travado com um valor de amplitude máximo e um valor de amplitude mínimo, que possam ser oriundos de algum erro de medida, é necessário reinicializar os valores de amplitude máxima e de amplitude mínima com os maiores valores numericamente possíveis. Desse modo, quando o sinal “Rising” vai a nível lógico “1”, o valor de amplitude máxima é reinicializado com o menor valor numericamente possível, e quando o sinal

“Failling” vai a nível lógico “1”, o valor de amplitude mínima é reinicializado com o maior valor numericamente possível. Os dois valores não podem ser reinicializados simultaneamente. Se isso ocorrer, o valor da componente CC será calculado de maneira errada, pois enquanto o valor da amplitude máxima estiver variando, o valor da amplitude mínima estará com um valor errado, e vice-versa.

Os blocos “Exceção_Valor_Max” e “Exceção_Valor_Min” tem a função de tratamento de erro. Esses blocos podem ser utilizados para limitar os valores máximos e mínimos que os valores de pico podem assumir na faixa de operação normal. No detector implementado, os blocos de tratamento de erro são utilizados apenas para enviar um valor de amplitude zero quando o maior valor numericamente possível e o menor valor numericamente possível são detectados. Essa ação foi tomada devido ao bloco “Scope” do Simulink criar uma escala com um valor máximo muito acima da faixa de operação do sinal de entrada mascarando a visualização dos resultados. O motivo desse mascaramento é que inicialmente o valor da máxima amplitude é o menor valor numericamente possível, e o valor da amplitude mínima é o maior valor numericamente possível. Após o primeiro ciclo do sinal de entrada ter sido completado, os valores no *buffer* de saída estarão todos dentro da faixa de operação de amplitude do sinal de entrada.

3.4 SIMULAÇÃO DA TÉCNICA DA FIGURA DE LISSAJOUS

Para a simulação da medição de fase através da técnica da figura de Lissajous, convencionou-se que o sinal de referência não tem componente CC e tanto o sinal de referência quanto o sinal defasado tem frequência de 50 kHz. O modelo Simulink utilizado para a simulação da medição através da técnica de Lissajous é mostrado na Figura 32.

Figura 32 - Medidor de fase pela técnica de Lissajous

O modelo implementado utiliza o detector de pico e componente CC (bloco “Detector_Pico_CC”) mostrado na Figura 31 para estimar o valor da componente CC do sinal defasado bem como o valor do seu pico máximo. O bloco “Detector_Pass_Zero” detecta quando o sinal de referência cruza o nível de tensão de 0 V, indo de um valor positivo para um negativo, e gera um pulso de *clock* para o bloco “Buffer_Valor_Va”, o qual armazena o valor do sinal defasado naquele instante de tempo.

Conforme a definição da equação (15), o valor do parâmetro b é obtido subtraindo-se o valor de pico máximo do sinal defasado, do valor da sua componente CC. Também, conforme a definição da equação (14), o parâmetro a é obtido, subtraindo-se do valor amostrado do sinal defasado, quando o valor de referência passa por zero (idealmente no instante que o sinal de referência é 0 volts), do valor da sua componente CC. A diferença de fase é obtida conforme a equação (10), calculando-se o arco-seno da razão entre parâmetro a e o parâmetro b .

Como o sinal defasado deve ser amostrado no instante que o sinal de referência é zero, a resposta exata livre de erros, é aquela obtida no exato instante de tempo no qual o sinal de referência é zero. Assim, considerando-se um sinal de referência senoidal expresso pela equação (119), a solução dessa equação, para o conjunto de instantes de tempo no qual o sinal de referência é zero, é expressa pela equação (120).

$$V_{REF}(t) = A \cdot \text{sen}(2\pi \cdot f \cdot t + \theta) \quad (119)$$

$$t = \frac{k - \theta/2\pi}{f}, \quad \text{para } k = 0, 1, 2, \dots, \infty \quad (120)$$

Pode-se definir um passo de tempo que inclua os instantes de tempo exatos nos quais o sinal de referência tem valor zero. Para isso, deve-se tomar o primeiro valor de k que resulta em um instante de tempo maior que zero (k_{t1}) na equação (120). Em seguida, substituindo-se a equação (120) na equação (117), obtém-se o valor mínimo que deve ter a constante β , resultando na equação (121).

$$\beta_{min} = \frac{1}{k_{t1} - \theta/2\pi} \quad (121)$$

Finalmente, substituindo-se a equação (121) na equação (118), obtém-se a equação (122), a qual fornece o valor do passo de tempo em segundos, para se obter uma simulação próxima do ideal, incluindo os instantes no qual o sinal de referência assume valor zero, respeitando o critério de Nyquist, e aumentando a quantidade dos pontos de discretização para se obter uma visualização melhor dos sinais simulados. Para a simulação da técnica da figura de Lissajous, a maior frequência presente é a do sinal de referência.

$$passo_{max} = \frac{k_{t1} - \theta/2\pi}{2 \cdot \beta \cdot 10^N} \cdot \frac{1}{frequencia} \quad (122)$$

Nas simulações executadas para o modelo da técnica da figura de Lissajous, foi utilizado o *solver* de passo variável ode45 (Dormand-Prince), o qual é considerado como tendo precisão média. O tempo final da simulação é o tempo equivalente ao final do décimo ciclo do sinal de referência ($t_F = 10 \cdot (1/f_{VREF})$). O número de amostras considerado dentro de um passo de tempo foi de 1000 ($N = 3$ na equação (122)). A fase do sinal defasado foi variada de 0° a 360° com passo de $0,1^\circ$. A defasagem medida pelo modelo é mostrada na Figura 33. Os valores calculados pelo modelo são comparados com o valor da função arco-seno teórico mostrado na Figura 27 e o erro de fase medida, em graus, é mostrado na Figura 34.

Figura 33 - Fase estimada pelo modelo Simulink para a técnica derivada da técnica da figura de Lissajous

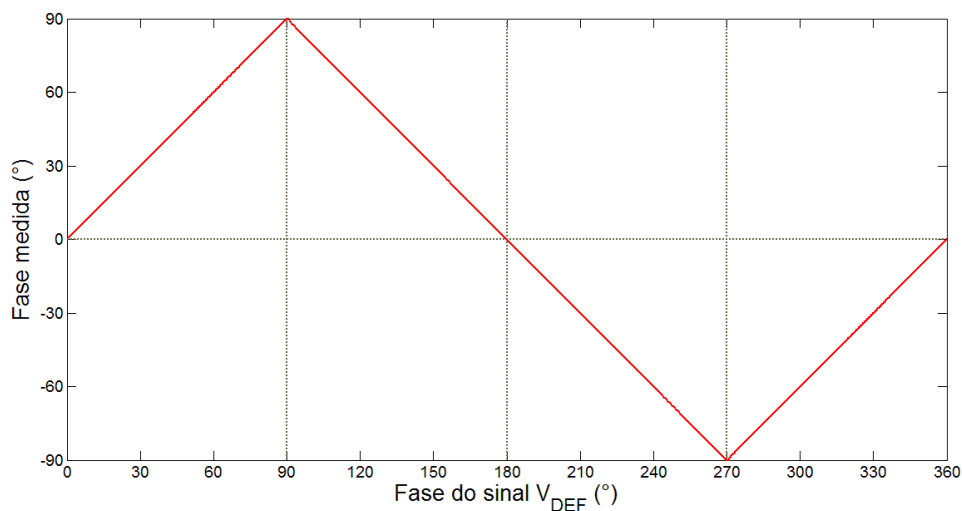
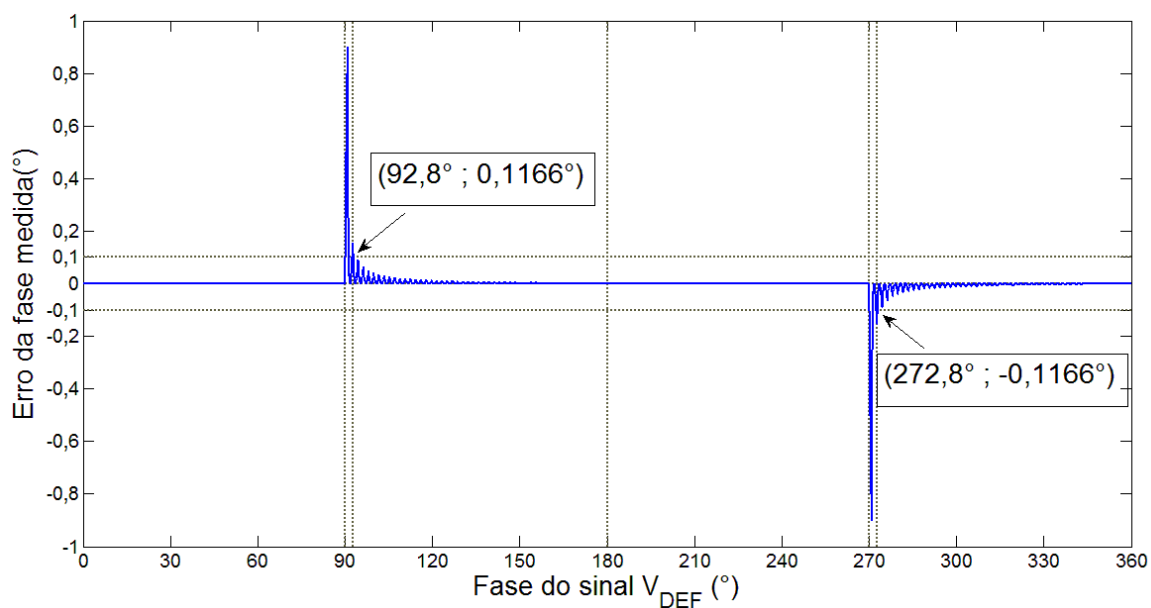


Figura 34 - Erro de estimativa de fase obtida pelo modelo Simulink para a técnica derivada da técnica da figura de Lissajous

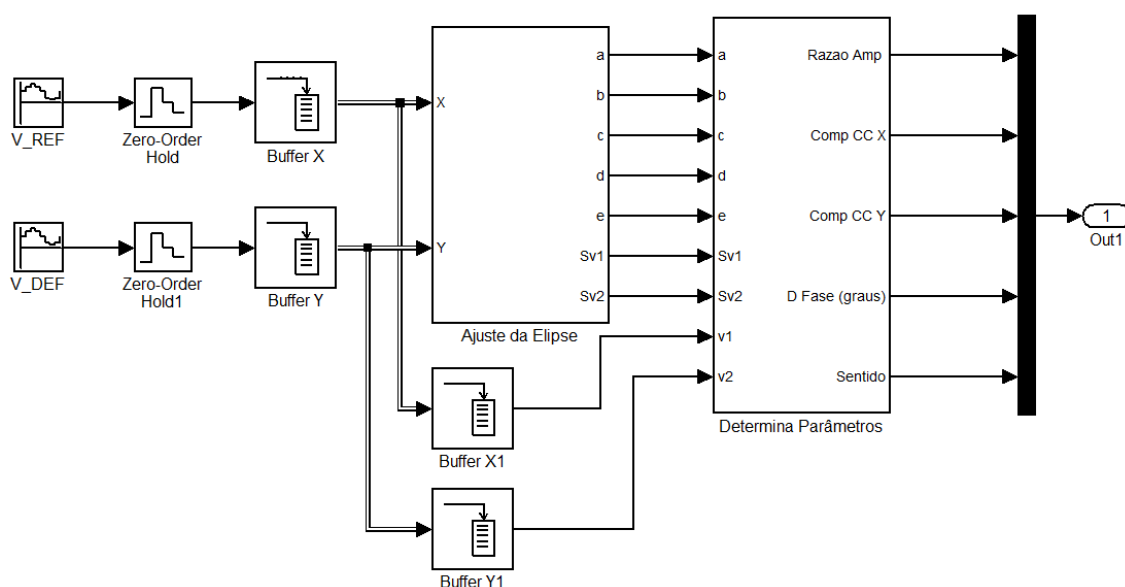


De acordo com os resultados obtidos para o modelo de determinação de defasagem pela técnica de Lissajous, o erro de fase, em módulo, foi menor que $0,1^\circ$ para a maior parte do intervalo testado, sendo maior que esse valor apenas nos intervalos $[90^\circ, 92,8^\circ]$ e $[270^\circ, 272,8^\circ]$.

3.5 SIMULAÇÃO DA TÉCNICA DO AJUSTE DA ELIPSE

O modelo utilizado para a medição de fase pela técnica do ajuste da elipse foi dividido em dois blocos fundamentais: um bloco com o algoritmo de ajuste da elipse proposto por Halir e Flusser (1998), e um bloco com as equações para a determinação dos parâmetros de razão de amplitude e diferença de fase conforme proposto por Ramos et al. (2009). A Figura 35 mostra o arranjo do primeiro nível mais alto do modelo (“*top level*”).

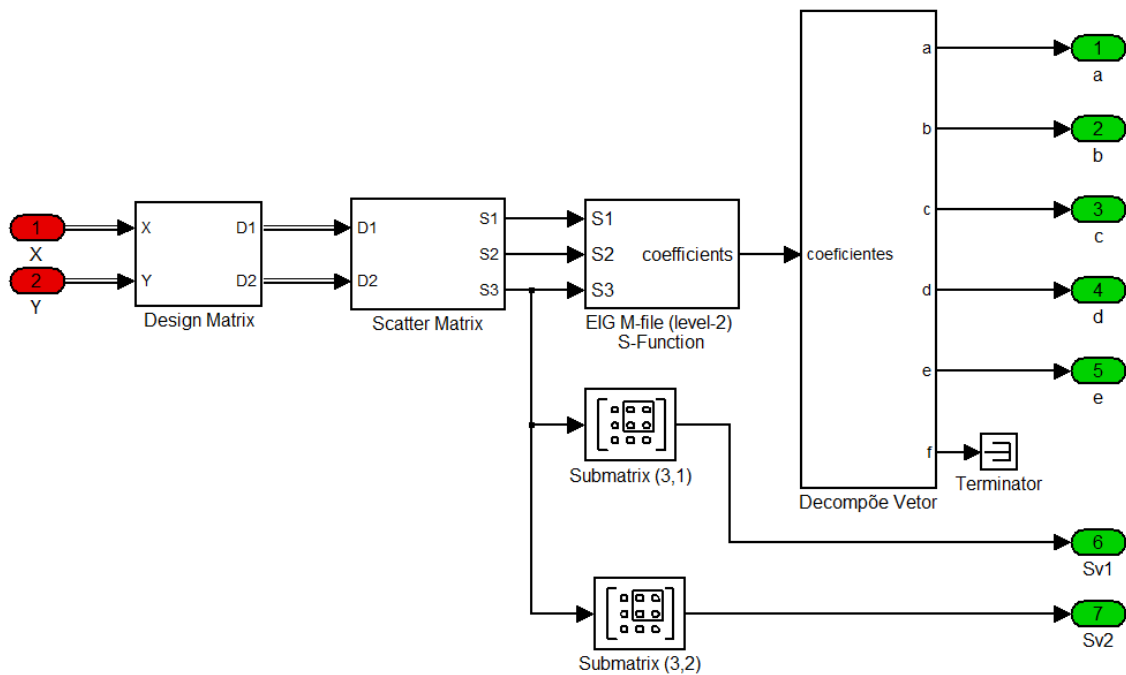
Figura 35 - Primeiro nível do modelo Simulink de medição de fase por ajuste da elipse



No “*top level*” do modelo, os blocos do tipo “Zero-Order Hold” servem para amostrar o sinal de referência e o sinal defasado em uma frequência de amostragem de 2 MHz. Os blocos “Buffer X” e “Buffer Y” funcionam com uma memória RAM que cria um vetor com 100 amostras, os quais repassados ainda na forma de vetor, para o bloco do algoritmo do ajuste da elipse. Os blocos “Buffer X1” e “Buffer Y1” servem para transformar os vetores em valores sequenciais. Esses valores são repassados para o bloco “Determina Parâmetros” e auxiliam na determinação do sentido de fase, ou seja, se a fase está adiantada ou atrasada. Os detalhes do bloco do algoritmo de ajuste de

elipse são mostrados na Figura 36 e os detalhes do bloco de determinação dos parâmetros de razão de amplitude e de diferença de fase são mostrados na Figura 38.

Figura 36 - Detalhes internos do bloco do algoritmo de ajuste da elipse



O bloco denominado “Design Matrix” cria as submatrizes D_1 e D_2 da matriz de design conforme as equações (30) e (31) (linhas 2 e 3 do algoritmo descrito na Figura 7). O bloco “Scatter Matrix” recebe as submatrizes de *design* e as utiliza conforme as equações (32) a (34) (linhas 4 a 6 do algoritmo) para criar as sub-matrizes de dispersão S_1 , S_2 e S_3 . As submatrizes de dispersão são então enviadas para o bloco denominado “EIF M-file (level-2) S-function” o qual resolve as equações (25) a (29) (linhas 7 a 13 do algoritmo Hali-Flusser) e gera o vetor com os coeficientes da elipse. Este bloco foi implementado através de *script* MATLAB (arquivo .m) pois existe a necessidade de tratamento de erro para a matriz M quando a submatriz de dispersão S_3 se aproxima de uma matriz singular. O tratamento é feito verificando-se se existe na matriz M algum número que representa infinito (“inf”) ou “not a number” (“NaN”). Caso exista algum desses dois indicadores, o bloco retorna um vetor de coeficientes com todos os

valores iguais a zero, e, caso não exista sinalização de erro, o algoritmo procede com a execução: calcula os autovalores e autovetores (linhas 10 e 11 do algoritmo Hali-Flusser), executa a verificação de qual autovetor resulta no menor autovalor positivo (linha 12 do algoritmo Hali-Flusser) e finalmente monta o vetor \vec{a} com os coeficientes da elipse (linha 13 do algoritmo Hali-Flusser). O código com a inserção do tratamento de erro poder ser visto na Figura 37. O bloco “Decompõe Vetor” simplesmente seleciona cada elemento do vetor de coeficientes separadamente, enviando-os para o bloco exterior “Determina Parâmetros”. O bloco “Submatrix (3,1)” seleciona o elemento S_{v1} da matriz S_3 e o bloco “Submatrix (3,2)” seleciona o elemento S_{v2} da matriz S_3 , ambos necessários para o cálculo do sentido da fase.

Figura 37 - Código MATLAB do bloco "EIG M-file (level-2) S-Function"

```
function Output(block)

    a=[0; 0; 0; 0; 0; 0];

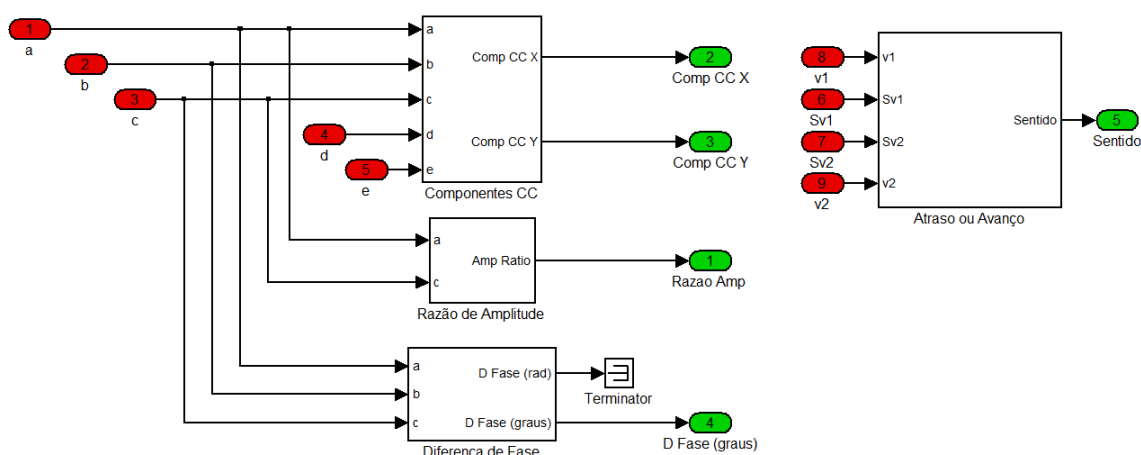
    S1=block.InputPort(1).Data;
    S2=block.InputPort(2).Data;
    S3=block.InputPort(3).Data;

    T = - inv(S3) * S2';
    M = S1 + S2 * T;
    M = [M(3, :) ./ 2; - M(2, :); M(1, :) ./ 2];

    Z=zeros(size(M));
    M_NAN=isnan(M);
    M_Inf=isinf(M);

    if ((M_NAN+M_Inf)==Z)
        [evec, eval] = eig(M);
        cond = 4 * evec(1, :) .* evec(3, :) - evec(2, :).^ 2;
        a1 = evec(:, find(cond > 0));
        a = [a1; T * a1];
    else
        a=[0; 0; 0; 0; 0; 0];
    end;

    block.OutputPort(1).Data = a;
```

Figura 38 - Diagrama interno do bloco "Determina Parâmetros"

O bloco denominado “Componentes CC” determina a componente CC do sinal de referência e a componente CC do sinal defasado através das equações (46) e (47). O bloco “Razão de amplitude” utiliza a equação (45) para determinar a razão de amplitude e o bloco “Diferença de Fase” utiliza a equação (48) para determinar a diferença de fase, e transforma o ângulo determinado com a função arco-cosseno em radianos para o ângulo equivalente em graus. O bloco denominado “Atraso ou Avanço” resolve as equações (49) a (52) determinando se a diferença de fase é referente a um sinal atrasado ou adiantado. Caso esteja atrasado, retorna valor +1 e caso esteja adiantado retorna valor -1. Quando a soma cumulativa dos votos se anula, o bloco retorna valor zero.

Tendo em vista que o método do ajuste da elipse não só determina a diferença de fase entre os dois sinais, mas também suas componentes CC, a razão de amplitude e o sentido da fase, as simulações foram feitas com o sinal de referência e com o sinal defasado, ambos possuindo uma componente CC, e com amplitudes diferentes. A frequência utilizada para ambos os sinais foi de 50 kHz. A equação (123) representa o sinal de referência e a equação (124) o sinal defasado.

$$V_{REF}(t) = 1 \cdot \text{sen}(2\pi \cdot f \cdot t + 0^\circ) + 1.5 \quad (123)$$

$$V_{DEF}(t) = 2 \cdot \text{sen}(2\pi \cdot f \cdot t + \phi) + 3 \quad (124)$$

O algoritmo da elipse tem a limitação de não conseguir calcular valores próximos de 0° e de 180° , deste modo, limitou-se o intervalo de simulação aos dois intervalos: $[1^\circ, 179^\circ]$ e $[181^\circ, 359^\circ]$. O *solver* utilizado foi o *solver* de passo variável ode45 (Dormand-Prince), com o tempo final de simulação equivalente a cinco ciclos do sinal de referência.

O vetor angular $[1^\circ, 45^\circ, 90^\circ, 135^\circ, 179^\circ, 181^\circ, 225^\circ, 270^\circ, 315^\circ, 359^\circ]$ foi testado com tempo final de simulação de 1 segundo e com tempo final de simulação de 5 ciclos do sinal de referência. Ambos os tempos finais de simulação apresentaram o mesmo resultado, deste modo, estabeleceu-se 5 ciclos do sinal de referência como o tempo final para se executar a simulação de todos os pontos dos intervalos: $[1^\circ, 179^\circ]$ e $[181^\circ, 359^\circ]$. O passo angular para a simulação foi de $0,1^\circ$. O passo de tempo máximo utilizado para o *solver* foi o mesmo utilizado para a técnica da figura de Lissajous, considerando mil pontos de amostragem dentro do passo de tempo.

A diferença de fase estimada pelo modelo é mostrada na Figura 39, o sentido da fase é mostrado na Figura 40 e o erro da diferença de fase, em graus, é mostrada na Figura 41.

Figura 39 - Fase estimada pelo modelo Simulink para a técnica derivada da técnica do ajuste de elipse

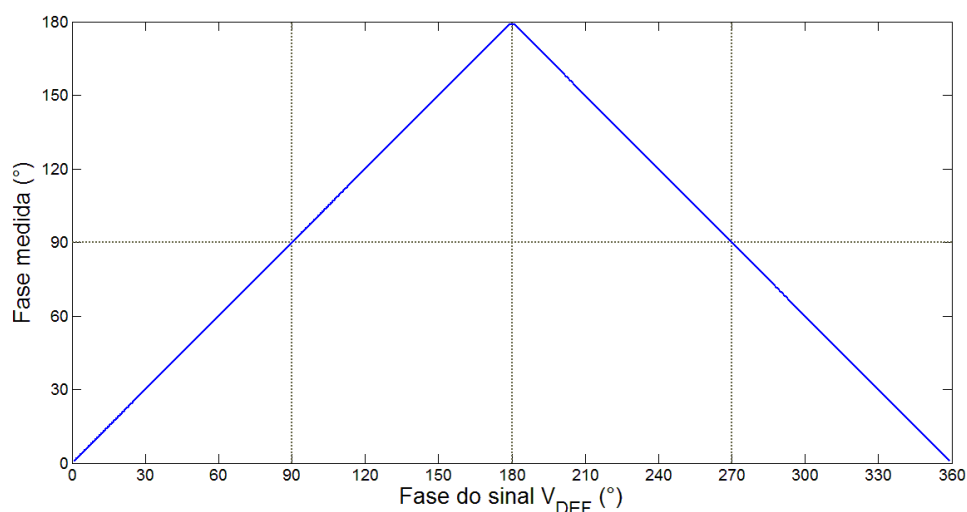


Figura 40 - Sentido de fase estimado pelo modelo Simulink para a técnica derivada da técnica do ajuste de elipse

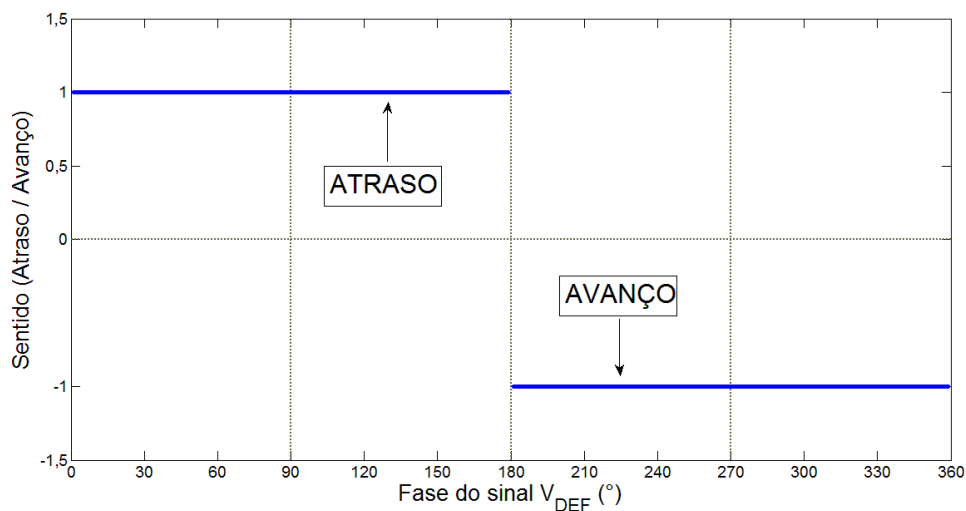
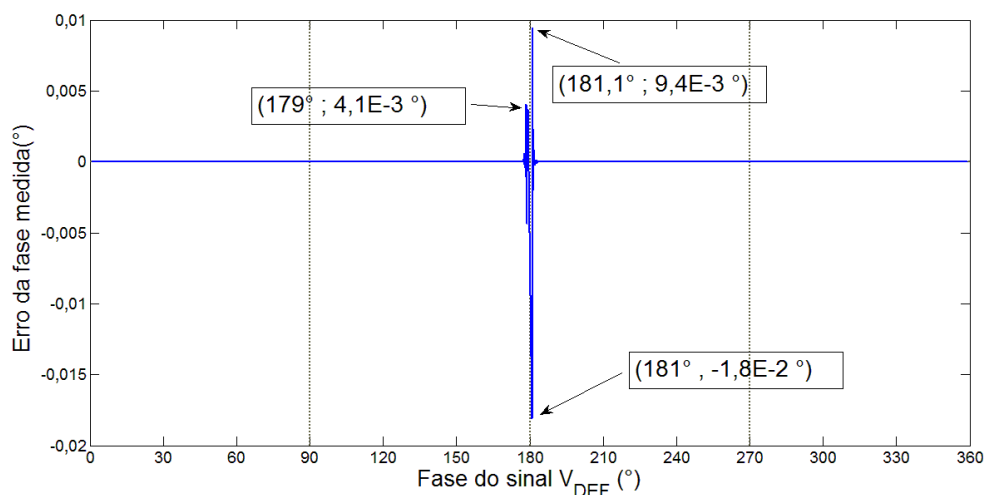


Figura 41 - Erro da fase estimada pelo modelo Simulink para a técnica derivada da técnica do ajuste de elipse



Observa-se na Figura 41 que o erro de fase é da ordem de centésimo de grau, apresentando picos nos ângulos entre 178° a $181,1^\circ$, o que é uma característica esperada para o algoritmo de ajuste da elipse. A Figura 40 mostra que a estimativa do sentido de fase foi correta, retornando valor +1 (atraso) para ângulos menores que 180° e valor -1 (avanço) para ângulos maiores de 180° . A fase estimada pelo modelo conforme mostrada na Figura 39 está de acordo com

a resposta da função arco-cosseno mostrada na Figura 29, indicando estimativa correta do ângulo de fase.

A Figura 42 e a Figura 43 mostram as componentes CC estimadas pelo modelo para o sinal de defasado e o sinal de referência, respectivamente.

Figura 42 - Componente CC do sinal defasado estimado pelo modelo Simulink para a técnica derivada da técnica do ajuste de elipse

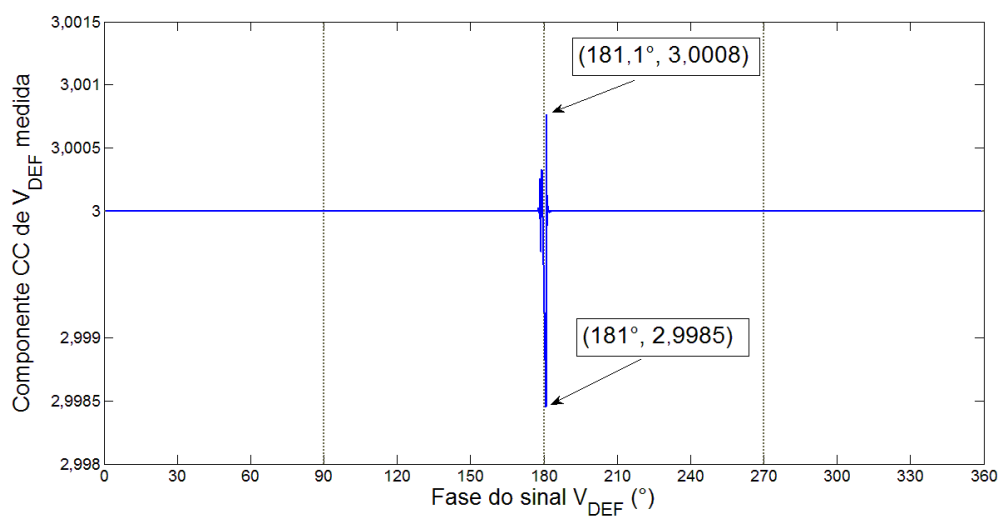
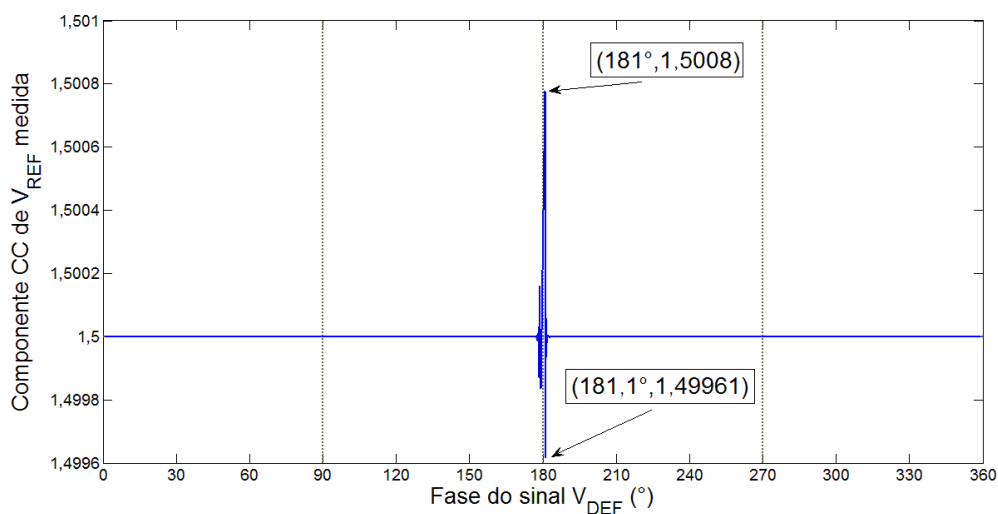


Figura 43 - Componente CC do sinal de referência estimado pelo modelo Simulink para a técnica derivada da técnica do ajuste de elipse

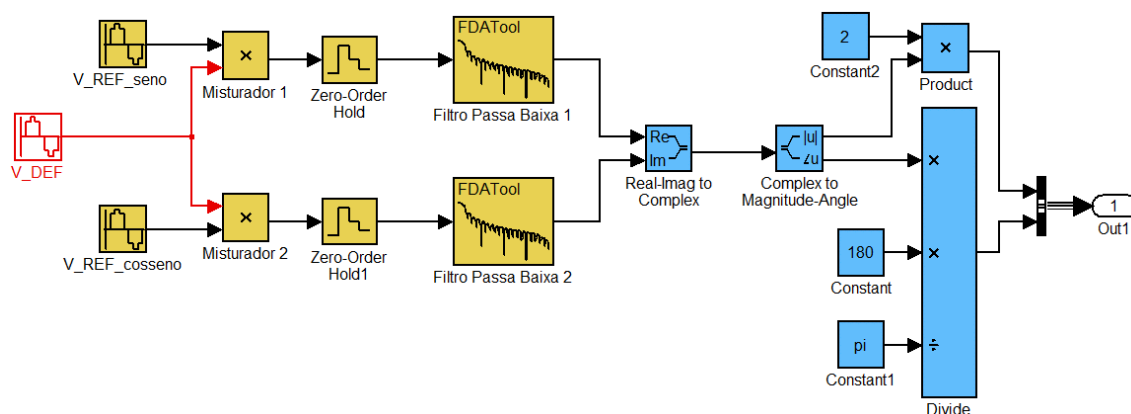


De acordo com os resultados apresentados na Figura 42 e na Figura 43, o valor numérico estimado para as componentes CC (tanto do sinal defasado quanto do sinal de referência), apresentaram uma oscilação numérica no intervalo entre 178° e 180° . Porém, essa oscilação é desprezível, estando dentro do erro de representação numérica.

3.6 SIMULAÇÃO DA TÉCNICA DE DEMODULAÇÃO DE QUADRATURA COM FILTRO FIR DIGITAL

As simulações da técnica de medição de fase pela demodulação de quadratura com filtro FIR digital foram executadas com o modelo mostrado na Figura 44. Os blocos com fundo bege são os blocos que compõe o demodulador de quadratura e os blocos com fundo azul são utilizados para calcular a fase e a amplitude do sinal defasado de acordo com as equações (70) e (71).

Figura 44 - Modelo Simulink para medição de fase através de demodulação de quadratura



Os blocos do tipo “Zero Order” amostram o sinal para ser enviado ao filtro e foram configurados com um tempo de amostragem equivalente ao período da frequência de 2 MHz. Os sinais de referência são de frequência 50 kHz. Os blocos “Filtro Passa Baixa” são filtros FIR do tipo digital e foram projetados com

frequência de corte de 1 KHz, frequência de amostragem de 2 MHz, pelo método da janela, e uma janela adicional passa banda do tipo Hanning.

O *solver* utilizado para a simulação foi o *solver* de passo variável e tempo discreto “discrete (*no continue states*)” e o tamanho máximo do passo de tempo foi configurado como automático. Utilizando-se o sinal defasado expresso na equação (126) como caso teste, foram testados os tempos de simulação: 1 segundo, 100 ciclos do sinal defasado e 10 ciclos do sinal defasado. Todos os tempos de simulação apresentaram o mesmo resultado estável, escolhendo-se então 10 ciclos do sinal defasado como o tempo final de simulação para os testes com o demodulador de quadratura. A equação (125) apresenta os parâmetros dos sinais utilizados como referência.

$$\begin{aligned} V_{REF_seno}(t) &= 1 \cdot \text{sen}(2\pi \cdot f \cdot t + 0^\circ) + 0 \\ V_{REF_cosseno}(t) &= 1 \cdot \cos(2\pi \cdot f \cdot t + 0^\circ) + 0 \end{aligned} \quad (125)$$

$$V_{DEF_padrão}(t) = 2 \cdot \text{sen}(2\pi \cdot f \cdot t + 30^\circ/180^\circ \cdot \pi) + 0 \quad (126)$$

O demodulador de quadratura tomado como referência mostrado por Martinez-Nolasco et al. (2012) utilizava um filtro FIR de ordem 51. Para verificar qual seria a ordem mais adequada ao filtro FIR a ser utilizado no demodulador de quadratura do modelo da Figura 44, observou-se a resposta de magnitude e de fase medidas, variando-se a ordem do filtro dentro do intervalo [2, 51]. A Figura 45 mostra a amplitude do sinal demodulado estimado pelo modelo para a variação da ordem do filtro, e a Figura 46 mostra a fase estimada.

Figura 45 - Amplitude estimada pelo demodulador de quadratura para a variação da ordem do filtro passa baixa FIR

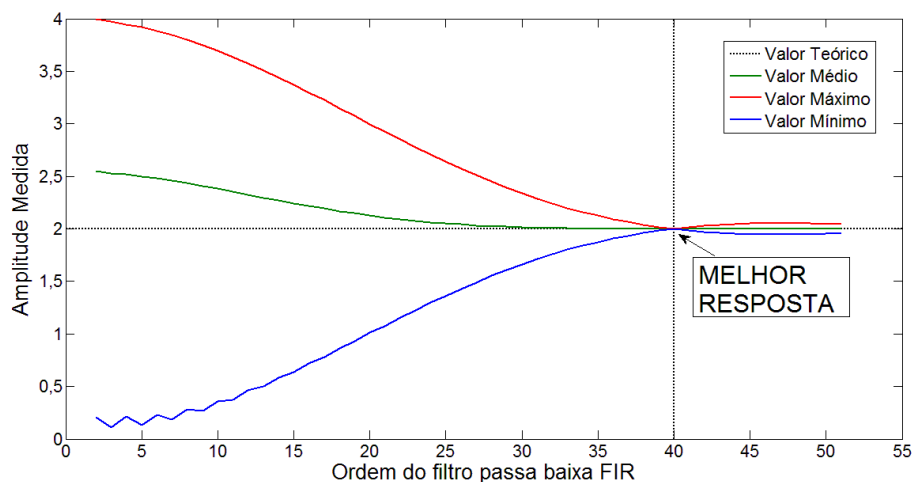
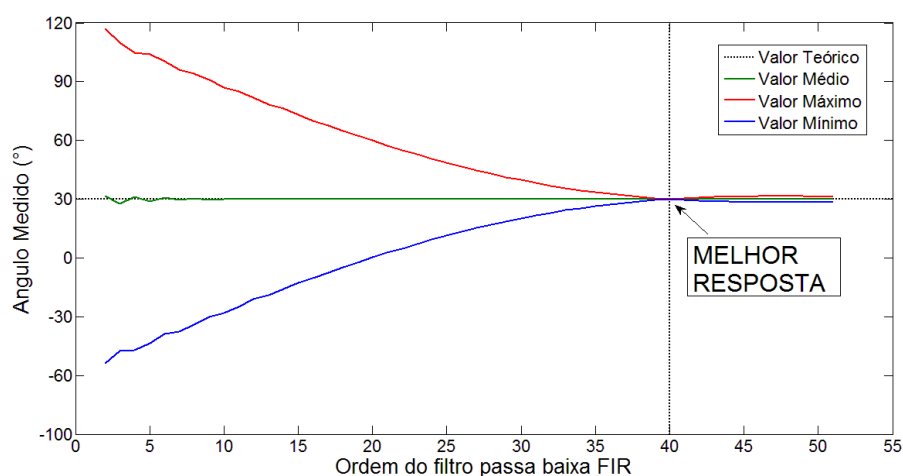


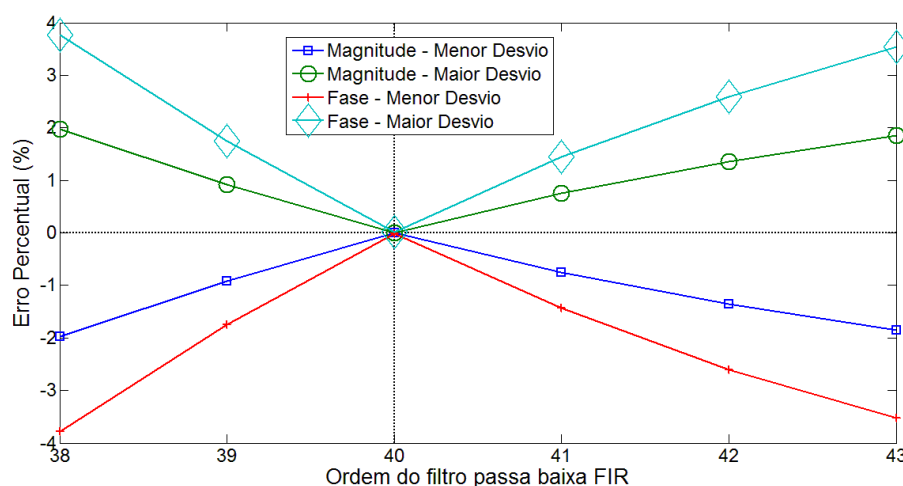
Figura 46 - Fase estimada pelo demodulador de quadratura para a variação da ordem do filtro passa baixa FIR



Os valores máximo e mínimo são obtidos com a média dos picos positivos e com a média dos picos negativos respectivamente. Através da resposta de magnitude e fase, verifica-se que a melhor resposta do filtro, ou seja, o ponto no qual o valor médio, o valor de desvio máximo e o valor de desvio mínimo são simultaneamente os mais próximos dos valores teóricos de magnitude e fase do sinal defasado, corresponde à ordem 40. A Figura 47 mostra o erro percentual entre os desvios máximos e mínimos em relação ao valor teórico de magnitude

e fase, para a faixa de ordem do filtro próxima a 40. Os erros zeram porque coincidentemente, para a configuração específica do filtro, ocorre um ponto ideal.

Figura 47 - Erro percentual dos desvios máximos e mínimos de magnitude e fase para ordem de filtro próxima a 40



Utilizando-se a ordem de melhor resposta (ordem 40) para os filtros do demodulador de quadratura, procedeu-se com a simulação variando-se a fase do sinal defasado no intervalo $[0^\circ, 360^\circ]$ com passo de $0,1^\circ$. A Figura 48 mostra a fase estimada pelo modelo e a Figura 49 o erro dessa estimativa em graus.

Figura 48 - Fase estimada pelo modelo Simulink pela técnica de demodulação de quadratura.

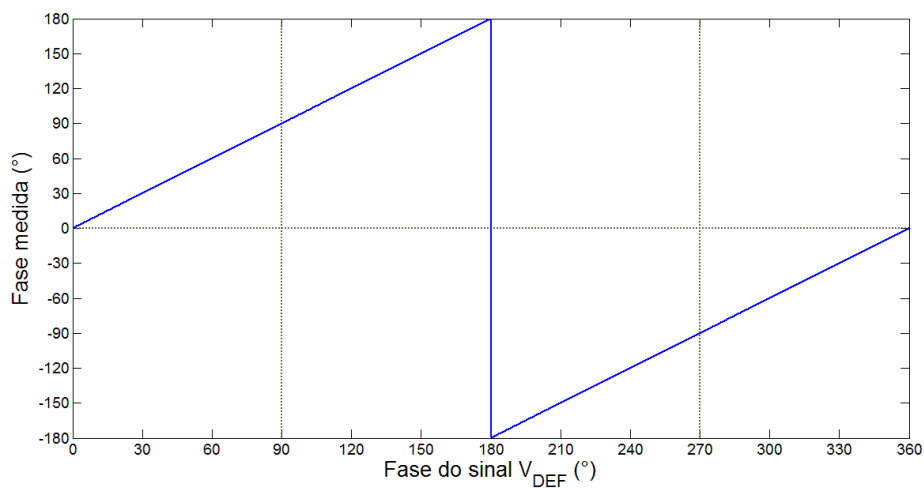
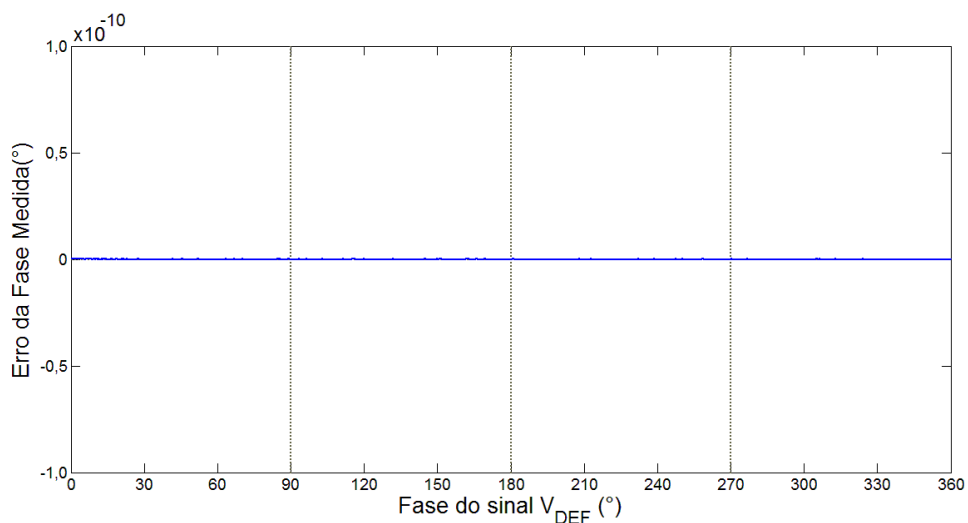


Figura 49 - Erro da fase estimada pelo modelo pela técnica da demodulação de quadratura



O bloco do modelo Simulink “Complex To Magnitude-Angle” utiliza uma função arco-tangente de quatro quadrantes, desta maneira, a fase medida pelo modelo se encontra dentro do intervalo $[-\pi, \pi]$. O erro de fase ficou equivalente ao erro numérico, podendo ser considerado zero e indicando que os resultados estimados para a fase são válidos. A Figura 50 mostra a magnitude do sinal defasado estimada e a Figura 51 mostra o erro percentual dessa estimativa.

Figura 50 - Magnitude estimada pelo modelo para a técnica da demodulação de quadratura

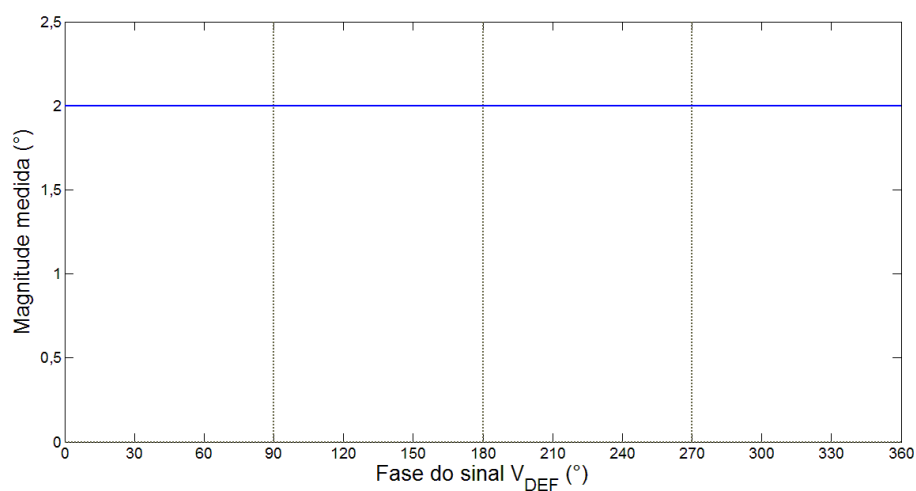
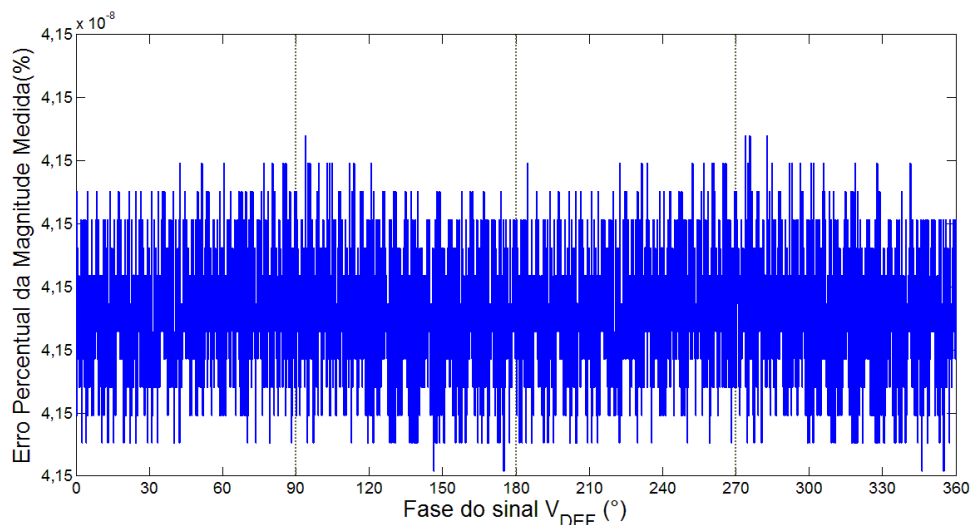


Figura 51 - Erro percentual para a estimativa de magnitude pela técnica da demodulação de quadratura



O erro percentual para a estimativa da magnitude tem a grandeza de ruído numérico indicando que o valor medido para a magnitude foi constante. O baixo erro encontrado tanto para a estimativa de fase (próximo do erro numérico) quanto para estimativa de magnitude ($4,15 \times 10^{-8} \%$) indicam que a ordem 40 é a ordem ideal para o filtro passa baixa utilizado, projetado com frequência de corte de 1 kHz e frequência de amostragem de 2 MHz.

3.7 SIMULAÇÃO DA TÉCNICA DE MEDIÇÃO DIRETA DIGITAL

Conforme descrito anteriormente, a medição de fase direta digital no conceito apresentado na Figura 11 não consegue medir a fase de um sinal que está adiantado. Para solucionar essa limitação foi implementada a alteração conceitual mostrada na Figura 52, a qual consiste na inserção de dois monoastáveis que são disparados por borda de subida. O pulso gerado pelos monoastáveis é de curtíssima duração, provocando a alteração do estado do *flip-flop* apenas no momento da borda de subida, fazendo com que o *set* do *flip-flop* seja comandado apenas na transição do semiciclo negativo para o semiciclo positivo do sinal defasado, e com que o *reset* do *flip-flop* seja comandado apenas

No modelo mostrado na Figura 53, os blocos “Gain” E “Compare To Zero” são os comparadores com zero que transformam o sinal senoidal em uma forma de onda digital. Quando a forma de onda digital correspondente ao sinal defasado provoca uma transição de baixo para alto na entrada do bloco “Monoastável START”, a saída do *flip-flop* RS vai para nível lógico “1”, fazendo com que o valor que estava armazenado internamente no bloco “Buffer de Saída” seja enviado para o sinal “Data Out”. Ainda, devido ao nível lógico “1” na saída do flip-flop RS, o contador de pulsos é zerado, o gerador de trem de pulsos de contagem é ativado e a operação *AND* permite que o trem de pulsos gerado incremente o contador. Quando a forma de onda digital correspondente ao sinal de referência provoca uma transição de alto para baixo na entrada do bloco “Monoastável STOP”, a saída do *flip-flop* RS vai a zero interrompendo, desativando o gerador de pulsos e fazendo com que a operação *AND* leve a entrada de *clock* do contador a nível lógico “0”.

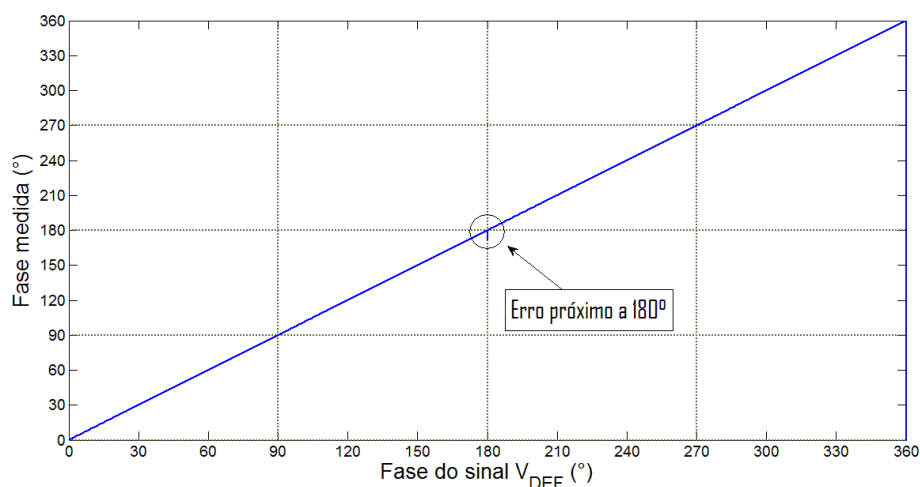
Para se definir o tempo final de simulação foi testado um caso padrão com os sinais definidos nas equações (127) e (128). Testou-se inicialmente 100 ciclos do sinal de referência e observou-se que a fase medida variou em $\pm 0,1^\circ$ ao longo do tempo testado. Dessa forma, foram testados tempos no intervalo $[0, 10^{-2}]$ segundos, com passo de 10^{-4} . Para cada tempo testado calculou-se a média e truncou-se os valores menores que a primeira casa decimal. Observou-se que o tempo final de 0,5 ms é suficiente para o sinal apresentar as oscilações de erro de $\pm 0,1^\circ$ e fornecer um resultado representativo da fase medida.

$$V_{REF_seno}(t) = 1 \cdot \text{sen}(2\pi \cdot f \cdot t + 0^\circ) + 0 \quad (127)$$

$$V_{DEF_padrão}(t) = 2 \cdot \text{sen}(2\pi \cdot f \cdot t + 30.3^\circ) + 0 \quad (128)$$

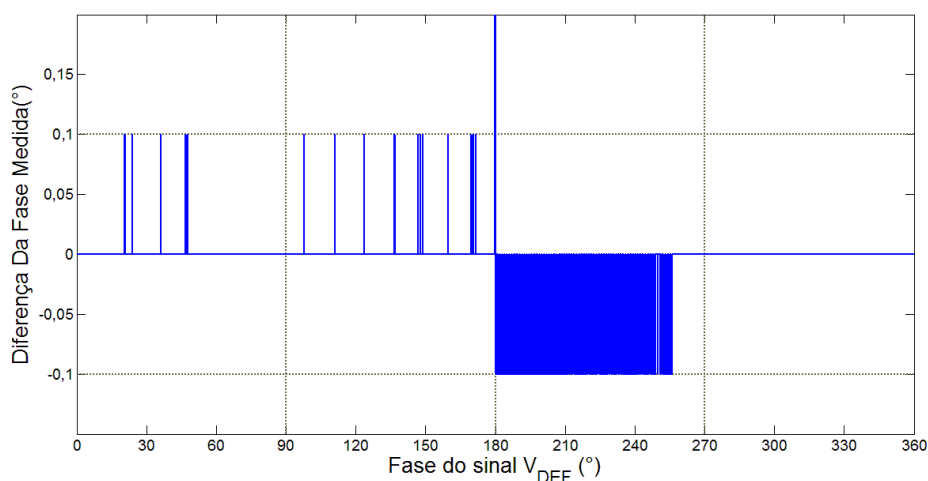
O *solver* utilizado para a simulação é o *solver* com passo variável ode45 “(Dormand-Prince)”, com o passo máximo de tempo configurado para automático, e tempo final de simulação de 0,5 ms. Para a simulação variou-se a fase do sinal defasado para todo o intervalo $[0^\circ, 360^\circ]$ com passo de $0,1^\circ$. A resposta de fase estimada pelo modelo é mostrada na Figura 54.

Figura 54 - Fase estimada pelo modelo de medição direta digital



Devido à contagem dos pulsos ser diretamente proporcional à fase do sinal defasado, na proporção de 1 pulso pelo valor de resolução angular de projeto, a resposta desse tipo de modelo será sempre uma reta de fase medida versus número de pulsos, sendo o coeficiente linear igual a zero e o coeficiente angular igual a resolução angular de projeto. A resposta mostrada na Figura 54 está adequada ao tipo de resposta esperado na saída do modelo, porém, para o ângulo de 180°, ocorre um erro de medida mais grosseiro, cujo valor retornado é de 172°. A Figura 55 mostra a diferença entre a fase medida e a fase teórica.

Figura 55 - Erro da estimativa de fase do modelo Simulink para a técnica de medição direta digital de fase.

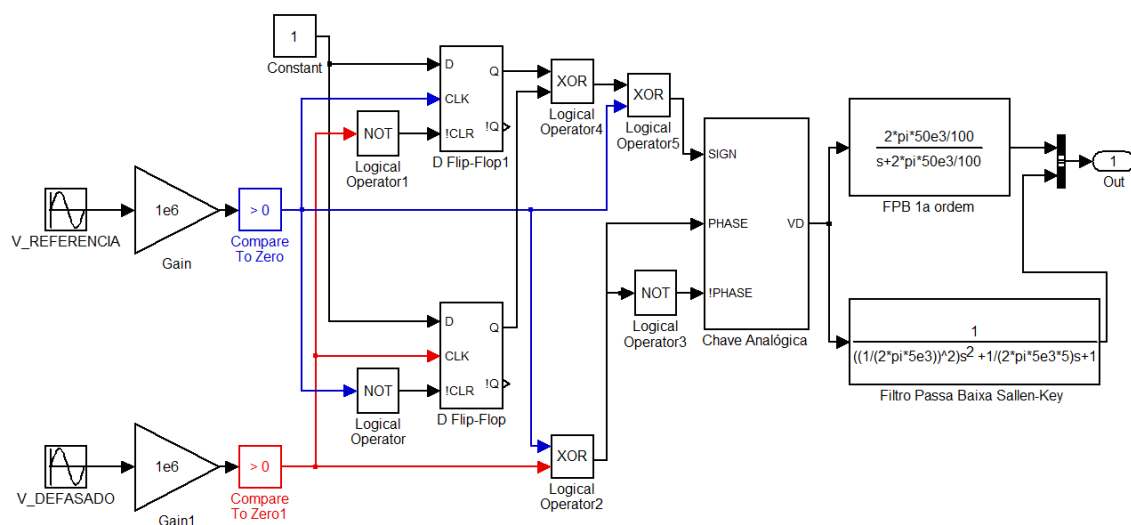


Exceto pelo intervalo de pontos numéricamente próximos de 180° (por exemplo: $[179,9999^\circ, 180,0001^\circ]$), todos os outros pontos mostraram uma variação máxima dentro da faixa de resolução de $\pm 0,1^\circ$. A razão dessa variação ocorrer é devido ao disparo dos pulsos de contagem. Se o disparo do *flip-flop* tanto para começar a contagem, ou para parar a contagem, ocorre simultaneamente com o pulso de *clock*, esse pulso pode ou não ser contado, levando ao erro de ± 1 pulso de contagem, que pela relação direta à resolução angular do modelo, leva ao erro de $\pm 0,1^\circ$. Valores superiores ou inferiores a essa variação são devido ao efeito cumulativo de se ter calculado a média dos valores de fase durante todo o período de simulação para representar o valor final.

3.8 SIMULAÇÃO DA TÉCNICA DE PWM

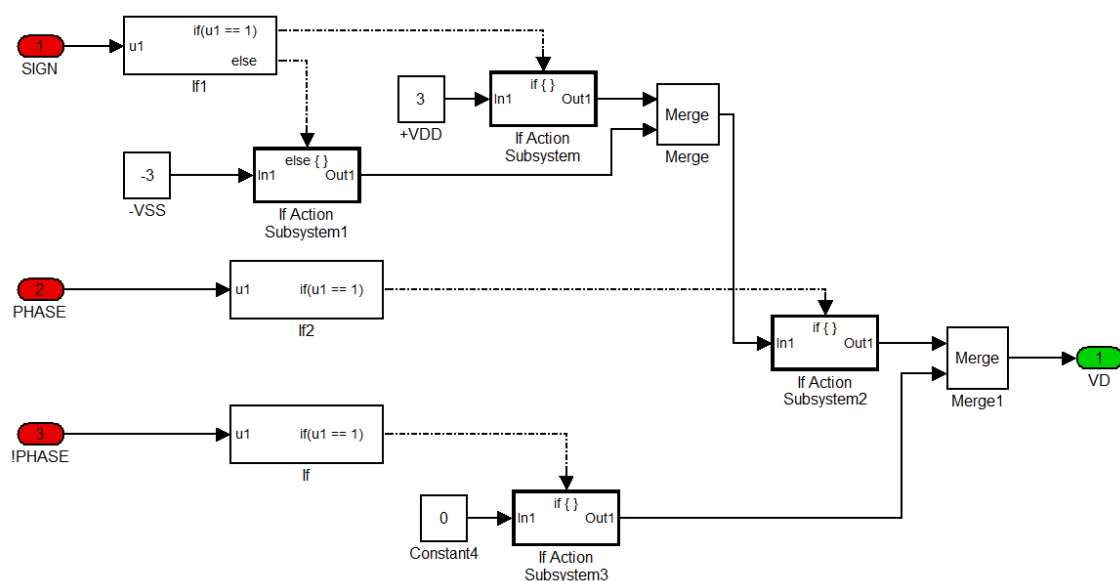
Para a simulação da técnica de medição de PWM foi implementado o modelo Simulink da topologia Lawson; Brown (1981), a qual foi inserida por Austin em um ASIC de tecnologia CMOS 350 nm. Foram testados o filtro passa baixa de primeira ordem e o filtro passa baixa de segunda-ordem de topologia Sallen-Key. O modelo Simulink é mostrado na Figura 56. Utilizando-se a relação da equação (78), para a frequência do PWM de 50 kHz, definiu-se o fator de redução de banda do filtro como sendo $K=100$ para o filtro de primeira ordem e $K=10$ para o filtro de segundo ordem. O tempo de resposta para valores maiores que 10 apresentou degradação na resposta do filtro, sendo que para valores da ordem de 100 o tempo de resposta do filtro é uma onda amortecida com período de amortecimento da ordem de segundos.

Figura 56- Modelo Simulink da topologia Lawson-Brown para a medição de diferença de fase através de PWM



Os blocos do tipo “Gain” e “Compare to Zero” formam os comparadores de entrada os quais transformam os sinais senoidais em formas de onda digitais. Toda a lógica da topologia digital foi implementada com valores booleanos de nível lógico “1” e nível lógico “0”, deixando-se a conversão dos níveis lógicos para níveis os de tensão $+VDD$ e $-VSS$ como uma função do bloco “Chave Analógica”, cuja topologia interna é mostrado na Figura 57.

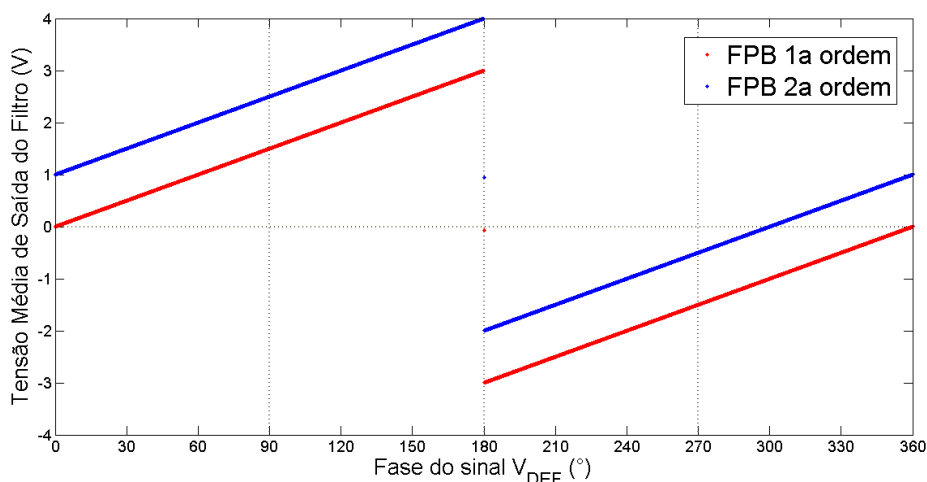
Figura 57 - Topologia interna do bloco "Chave Analógica" do modelo Simulink da topologia Lawson-Brown.



Utilizando-se o sinal de referência padrão da equação (127) e o sinal defasado padrão da equação (128), testaram-se vários tempos de simulação e passo máximo de tempo para o *solver*. O objetivo do teste foi determinar o tempo necessário para que a resposta dos filtros passa-baixa do modelo se estabilizassem em regime permanente. Dado aos resultados obtidos, optou-se pela utilização do *solver* de passo variável ode45 “(Dormand-Prince)” com passo máximo de tempo configurado como $1/(2 \cdot 10^2 \cdot 50 \times 10^3)$, e tempo final de simulação equivalente a 1000 ciclos do sinal de referência. Para a simulação, variou-se a fase do sinal defasado por todo o intervalo $[0^\circ, 360^\circ]$ com passo de $0,1^\circ$.

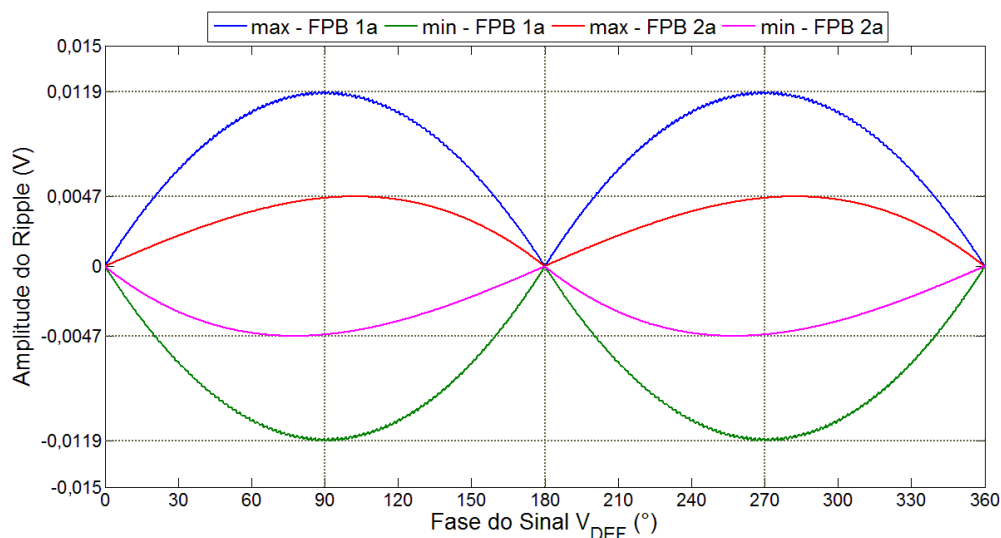
A saída do filtro apresenta *ripple*, desta forma, considerando-se todos os pontos da simulação, os quais representam o regime permanente (tempo maior que o tempo necessário para 200 ciclos do sinal de referência), foi determinada a tensão média de saída como sendo a resposta do modelo. A curva de tensão média obtida na saída dos filtros é mostrada na Figura 58. Os valores de tensão média obtidos nos dois filtros foram praticamente iguais, sendo a tensão média na saída do filtro de segundo ordem plotada com um *offset* positivo de 1 volt para facilitar a visualização.

Figura 58 - Tensão média na saída dos filtros



A Figura 59 mostra as tensões de pico para o *ripple* encontrado na saída dos filtros de primeira e segunda ordem, para cada fase do sinal defasado.

Figura 59 - Variação da amplitude da tensão de *ripple* na saída dos filtros



Reescrevendo o intervalo de fases original, que era $[0^\circ, 360^\circ]$, para o intervalo $[-180^\circ, +180^\circ]$, pode-se ajustar uma reta relacionando a tensão na saída do filtro com a diferença de fase. A Figura 60 mostra a reta ajustada para a saída do filtro de primeira ordem em função da fase do sinal defasado bem como a reta ajustada para se determinar a fase do sinal V_{DEF} em função da tensão na saída do filtro. A Figura 61 mostra as retas ajustadas para o filtro de segunda ordem.

Figura 60 - Retas ajustadas para o filtro de primeira ordem: a) Tensão como função da fase; b) Fase como função da tensão.

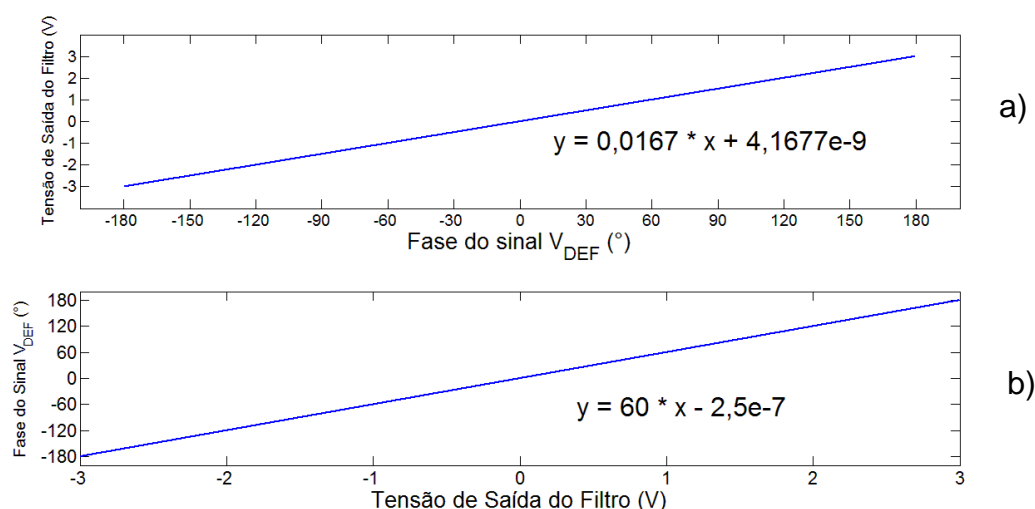
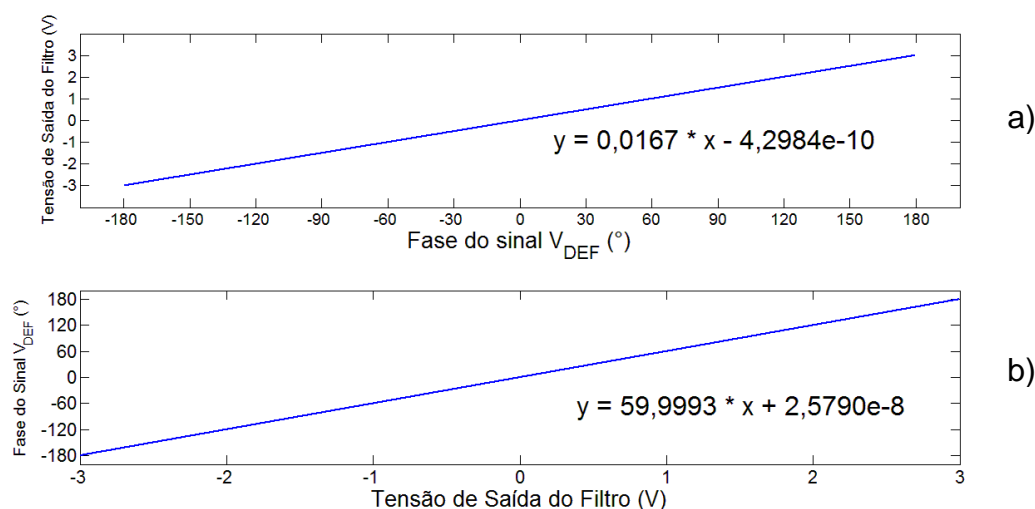
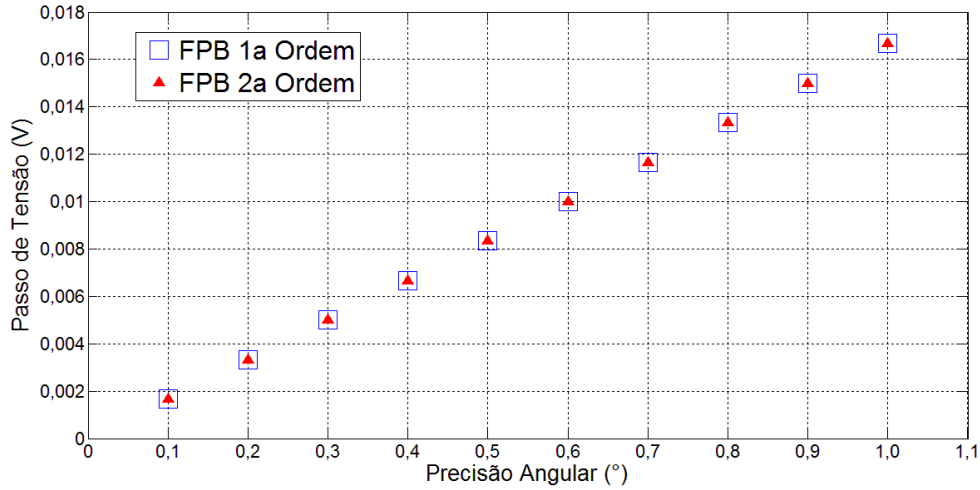


Figura 61 - Retas ajustadas para o filtro de segunda ordem: a) Tensão como função da fase; b) Fase como função da tensão.



As retas ajustadas para diferença de fase versus tensão média de saída do filtro, permitem se determinar qual é a discretização de tensão para a precisão angular desejada. Como as retas ajustadas resultam em um valor de 0 V para o ângulo de 0°, os valores de resolução desejados podem ser substituídos diretamente nas retas. A Figura 62 mostra a discretização de tensão média correspondente ao intervalo de resolução angular $[0,1^\circ, 1^\circ]$ com valores estimados a cada $0,1^\circ$.

Figura 62 - Pontos de discretização da tensão para o intervalo de resolução angular $[0,1^\circ, 1^\circ]$ com passo de $0,1^\circ$



Para se obter precisão na medida de fase, o valor da discretização de tensão correspondente à resolução angular utilizada, deve ser tal, que, tanto o maior valor de pico positivo do *ripple* quanto o menor valor de pico negativo do *ripple*, não provoquem um valor que indique um valor angular diferente da tensão média na saída do filtro. Dessa forma, o módulo da tensão de *ripple* tem que ser significativamente menor que o módulo da tensão média na saída do filtro, sendo idealmente expressa de acordo com o mostrado da equação (129), na qual \emptyset é o ângulo da fase medida e α é uma constante de proporcionalidade.

$$V_{discretização}(\emptyset) = \alpha \cdot V_{O_média}(\emptyset) \gg V_{O_média}(\emptyset) \pm V_{P_ripple} \quad (129)$$

Na condição limítrofe o valor da constante de proporcionalidade α deve impor uma tensão de discretização que não seja influenciada pelo *ripple*. Logo, sendo \emptyset o ângulo de fase medido e φ a resolução angular, a tensão média V_m na saída do filtro deve satisfazer simultaneamente as inequações (130) e (131).

$$V_m(\emptyset) + |V_{P_ripple_pos}(\emptyset)| < V_m(\emptyset + \varphi) - |V_{P_ripple_neg}(\emptyset + \varphi)| \quad (130)$$

$$V_m(\emptyset) - |V_{P_ripple_neg}(\emptyset)| > V_m(\emptyset - \varphi) + |V_{P_ripple_pos}(\emptyset - \varphi)| \quad (131)$$

De acordo com as curvas da Figura 59, a amplitude máxima do *ripple* varia de acordo com o ângulo de fase do sinal defasado. Rearranjando as inequações (130) e (131), obtém-se as inequações (132) e (133).

$$V_m(\emptyset + \varphi) - V_m(\emptyset) > |V_{P_ripple_pos}(\emptyset)| + |V_{P_ripple_neg}(\emptyset + \varphi)| \quad (132)$$

$$V_m(\emptyset) - V_m(\emptyset - \varphi) > |V_{P_ripple_neg}(\emptyset)| + |V_{P_ripple_pos}(\emptyset - \varphi)| \quad (133)$$

Tomando-se os valores do pior caso, considera-se o maior valor em módulo do pico positivo e o maior valor em módulo do pico negativo. Sendo o ângulo de fase zero graus equivalente a 0 V conforme observado pelas retas ajustadas para os filtros, a equação (132) resulta na equação (134) para os valores máximos de *ripple* do filtro de primeira ordem, e na equação (135) para os valores máximos de *ripple* do filtro de segunda ordem.

$$V_{m_FPB_1}(\varphi) > |V_{P_ripple_pos}| + |V_{P_ripple_neg}| > 0,0238 \quad (134)$$

$$V_{m_FPB_2}(\varphi) > |V_{P_ripple_pos}| + |V_{P_ripple_neg}| > 0,0095 \quad (135)$$

Substituindo-se os limites de tensão obtidos com as equações (134) e (135) na equação das retas ajustadas para os filtros, obtém-se o valor mínimo de resolução angular possível de se obter com o filtro simulado. Efetuando-se o cálculo, o valor de resolução máxima obtido com o filtro de primeira ordem é de $\varphi \approx 1,43^\circ$ e o valor de resolução máxima obtido com o filtro de segunda ordem é de $\varphi \approx 0,57^\circ$.

Caso o método PWM seja implementado em conjunto com uma solução analógica de filtro, é necessária a aquisição do sinal da saída do filtro com um ADC, e uma posterior conversão dos valores discretizados da saída do ADC em valores numéricos compatíveis com a resolução angular desejada. Devido ao *ripple* existente na saída do filtro, o cálculo de um valor médio de tensão com base em um número mínimo de amostras obtidas pelo ADC, pode levar a uma melhoria significativa do resultado. De forma semelhante, a substituição do filtro

analógico por um filtro digital, permite que um número mínimo de amostras obtidas diretamente na saída do filtro, possam ser utilizadas para o cálculo da tensão média.

3.9 SIMULAÇÃO DA TÉCNICA DE DFT

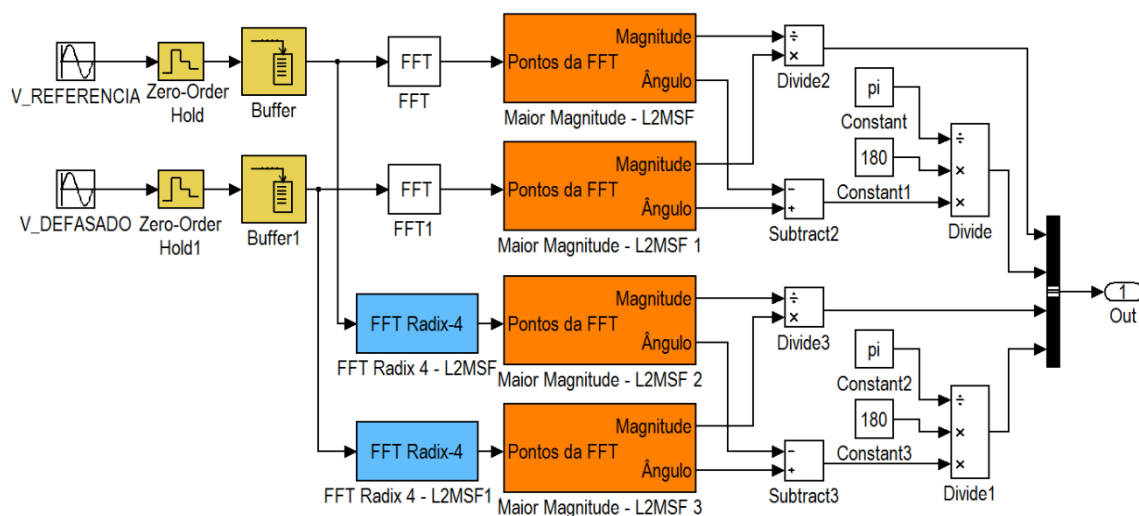
A técnica de medição de DFT foi implementada utilizando-se o bloco de FFT nativo do Simulink, o qual resolve a FFT através do algoritmo Radix-2, e um bloco de função S-Level tipo 2 no qual foi inserido o algoritmo Radix-4. O algoritmo Radix-4 utilizado foi a versão MATLAB do algoritmo para DSP apresentado por Zeev et al. (2007), cuja versão MATLAB foi disponibilizada por Donovan e Romero (2008) e reproduzida no Anexo A.

O número de pontos de amostragem a serem tomados para o algoritmo radix-4 deve ser sempre uma potência de 4^N , o que implica em uma frequência de amostragem definida também, conforme expresso pela equação (136), como uma potência de 4^N . O expoente N é um valor arbitrário.

$$f_{amostragem} = 4^N \cdot f_{sinal} \quad (136)$$

O modelo implementado no Simulink é mostrado na Figura 63. Os blocos com do tipo “Zero-Order Hold” e “Buffer”, realizam a função de amostragem e armazenamento de dados. Os blocos do tipo “FFT Radix 4 – L2MSF”, executam o algoritmo Radix-4 para FFT.

Figura 63 - Modelo Simulink para o teste da técnica de medição de razão magnitude e fase através da técnica de DFT



Os blocos do tipo “Maior Magnitude” executam a função de detectar qual é o elemento no resultado da FFT que tem a maior magnitude, retornando a magnitude e o ângulo desse elemento complexo. Esse bloco foi implementado através de *script* MATLAB associado a um bloco de função *S-Level* tipo 2. O trecho do código responsável por identificar o elemento de maior magnitude e retornar seus parâmetros na forma polar é mostrado na Figura 64.

Figura 64 - Trecho do código responsável pela funcionalidade dos blocos do tipo "Maior Magnitude - L2MSF"

```

62 function Output(block)
63
64     %Recebe vetor de elementos da FFT
65     X = block.InputPort(1).Data;
66
67     %Retorna magnitude e índice do elemento
68     %de maior magnitude
69     [mag_x idx_x] = max(abs(X));
70
71     %Retorna o angulo do elemento de maior
72     %magnitude
73     px = angle(X(idx_x));
74
75     %Envia valores na saída para a porta de saída
76     block.OutputPort(1).Data = mag_x;
77     block.OutputPort(2).Data = px;

```

O bloco FFT é o bloco nativo do Simulink que executa a FFT pelo algoritmo Radix-2. Os demais blocos são responsáveis pelo cálculo matemático para determinação da diferença de fase e razão de amplitudes conforme as equações (107) e (106). Também, executam a conversão de radiano para graus do ângulo de fase medido.

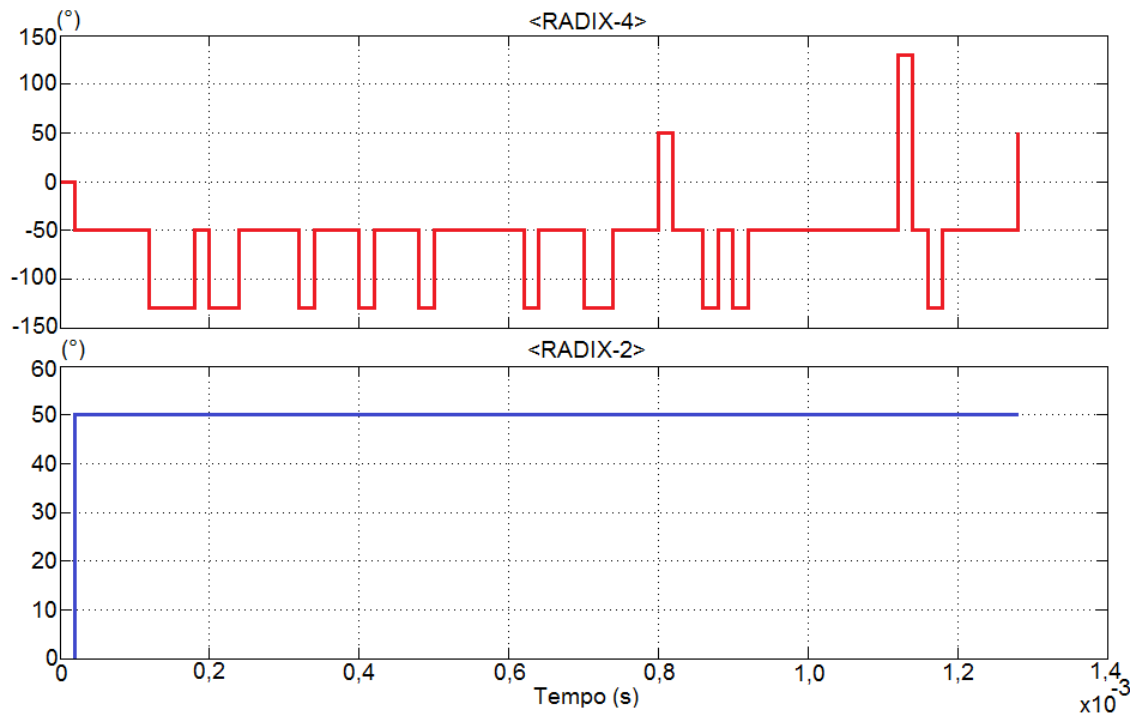
Conforme a implementação de FFT Radix-4 em FPGA mostrada por Bonilla et al. (2012), utilizou-se no modelo uma potência de 4^5 para a definição da frequência de amostragem, resultando em 1024 pontos de amostragem. A frequência de amostragem foi definida para um sinal de frequência de 50 kHz, o que de acordo com a equação (136), equivale a uma frequência de 51,2 MHz. O sinal de referência utilizado para as simulações obedece à função expressa na equação (137) e o sinal defasado obedece à função expressa na equação (138).

$$V_{REFERÊNCIA}(t) = 1 \cdot \text{sen}(2\pi \cdot f \cdot t + 0^\circ) + 0 \quad (137)$$

$$V_{DEFASADO}(t) = 2 \cdot \text{sen}(2\pi \cdot f \cdot t + \emptyset) + 0 \quad (138)$$

Para verificação inicial do algoritmo Radix-4, variou-se no modelo Simulink a fase do sinal defasado no intervalo $[0^\circ, 360^\circ]$ com passo de $0,1^\circ$, comparando-se a resposta do algoritmo Radix-4 frente à resposta do algoritmo Radix-2 da função FFT nativa do MATLAB. Observou-se que a resposta do algoritmo radix-4 falha em interpretar o quadrante correto do ângulo de fase, oscilando de forma a retornar o valor que o ângulo de fase teria em todos os quadrantes. A Figura 65 mostra a resposta do modelo Simulink para um sinal de defasagem com fase 50° .

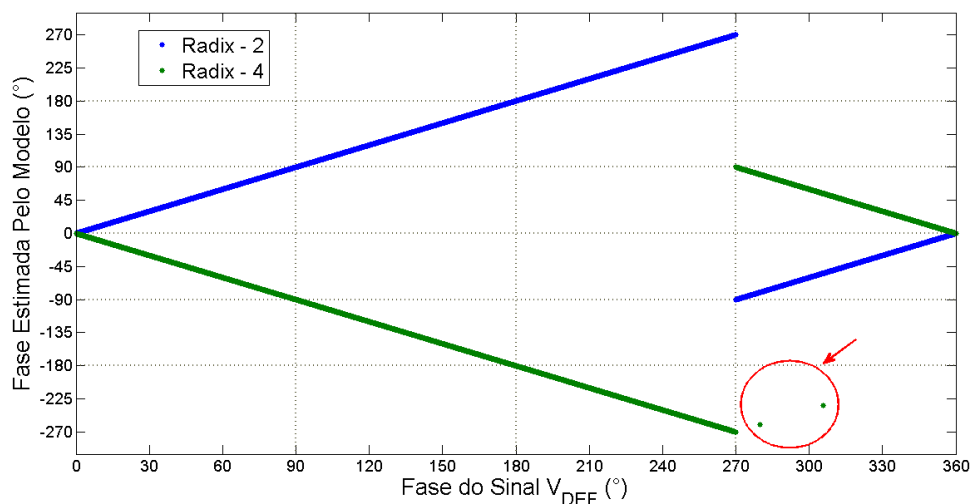
Figura 65 - Comparação da resposta entre o algoritmo Radix-4 e o algoritmo Radix-2 para o ângulo de 50°



Enquanto o algoritmo Radix-2 retorna o valor de 50° em regime permanente, o algoritmo Radix-4 retorna 50° (quadrante 1), -50° (quadrante 4), 140° (quadrante 2), e -140° (quadrante 3). Para se poder comparar a resposta do algoritmo Radix-4, os resultados da simulação devem ser tratados com a função *mode*, a qual retorna o valor que mais se repete em um conjunto de amostras.

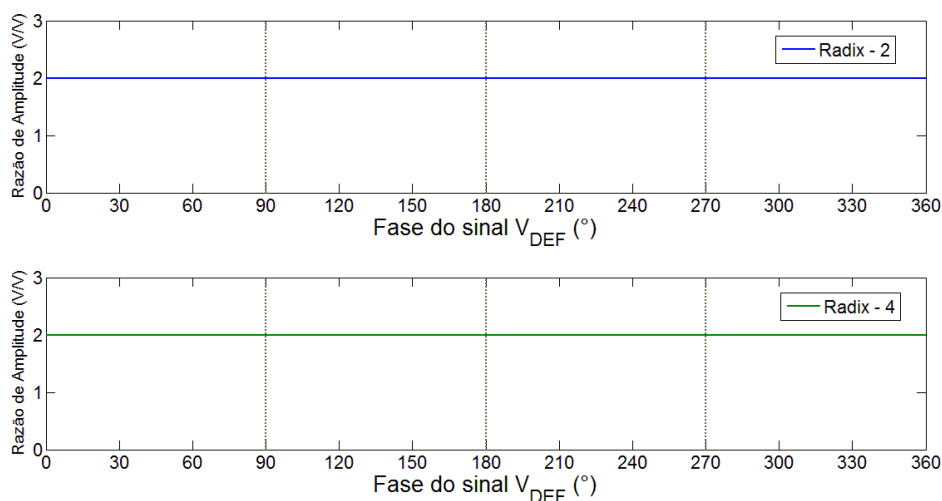
Os parâmetros de simulação foram definidos com o tempo final de simulação de 256 ciclos do sinal defasado, utilizando-se o *solver* de passo variável “discrete (no continuous states)”, com passo máximo de tempo automático. O valor da fase do sinal de referência foi variado para todo o intervalo $[0^\circ, 360^\circ]$ com passo de $0,1^\circ$ graus. A Figura 66 mostra a diferença de fase estimada pelo modelo e a Figura 67 mostra a razão de amplitudes obtida.

Figura 66 - Fase estimada pelo modelo FFT para 1024 pontos de amostragem



Através do resultado obtido para a resposta de fase, observa-se que o algoritmo Radix-4 processa os pontos amostrados no sentido horário do círculo trigonométrico, resultando em valores de fase negativos. Apenas dois pontos foram estimados de forma errada, indicando erro de resposta muito baixo.

Figura 67 - Razão de amplitudes estimada pelo modelo FFT para 1024 pontos de amostragem



A razão de amplitudes estimada pelo modelo não tem pontos medidos erroneamente, apontando que o algoritmo Radix-4 testado apresenta problemas relacionados a determinação dos “*twiddle factors*” durante a divisão do conjunto de amostras nos quatro subgrupos que são enviados para a operação da borboleta. A verificação da sensibilidade do algoritmo Radix-4 em relação ao número de pontos de amostragem é mostrada na Figura 68, na Figura 69 e na Figura 70 nas quais foram amostrados 256, 64 e 16 pontos respectivamente.

Figura 68 - Fase estimada pelo modelo FFT para 256 pontos de amostragem

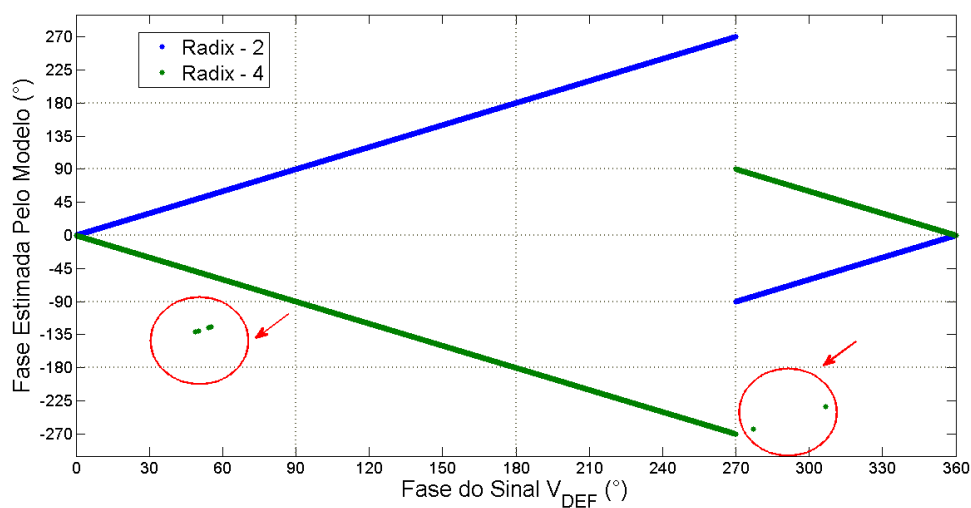


Figura 69 - Fase estimada pelo modelo FFT para 64 pontos de amostragem

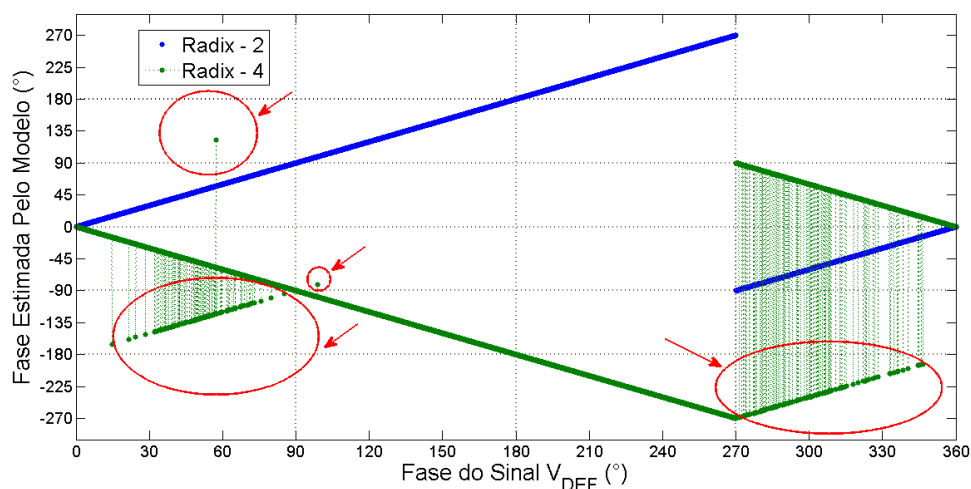
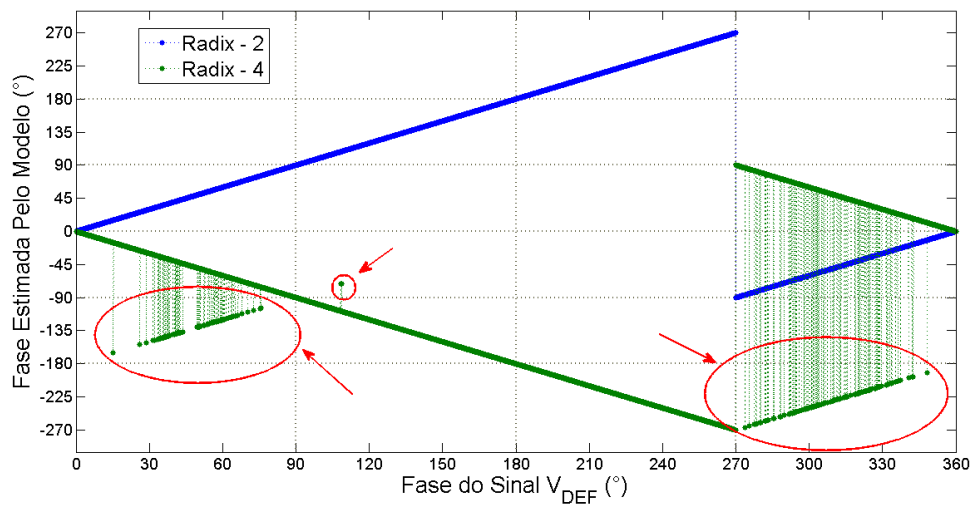


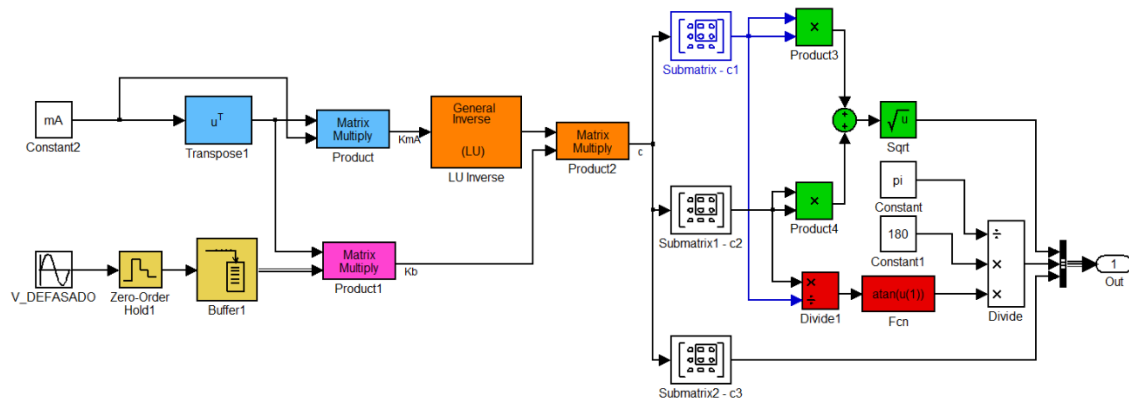
Figura 70 - Fase estimada pelo modelo FFT para 16 pontos de amostragem

Observa-se uma resposta pouco degradada para a amostragem com 256 pontos, aparecendo apenas 5 pontos medidos erroneamente. A utilização de 64 pontos e 16 pontos de amostragem apresentaram degradação semelhante entre si, tendo um péssimo resultado na estimativa da fase para ângulos nos intervalos $[0^\circ, 90^\circ]$ e $[270^\circ, 360^\circ]$.

3.10 SIMULAÇÃO DA TÉCNICA DO AJUSTE DA SENÓIDE

O modelo Simulink utilizado para a simulação da técnica de medição de amplitude e de fase pela técnica do ajuste da senóide é mostrado na Figura 71. O bloco “Constant2” carrega o valor da matriz \mathbf{mA} , cujos elementos são calculados através do código de script MATLAB mostrado na Figura 72, o qual utiliza a relação $\Omega = 2 \cdot \pi \cdot f_{V_DEFASADO} \cdot T_{amostragem}$, para preencher a matriz \mathbf{mA} com senos e cossenos de acordo com a definição da equação (110).

Figura 71 - Modelo Simulink para medição de fase e magnitude através da técnica do ajuste da senóide



O bloco “Zero-Order Hold1” e “Buffer1”, de fundo bege, são responsáveis por amostrar o sinal e memorizar o vetor \vec{b} composto por N amostras. Os blocos de fundo azul claro executam a equação (111) para criar a matriz KmA , o bloco de fundo rosa executa a equação (112) para criar o vetor \overline{Kb} e os blocos de fundo laranja executam a equação (113) para calcular o vetor \vec{c} . Os blocos de tipo “Submatrix” selecionam os elementos do vetor \vec{c} separadamente, para o cálculo do valor da amplitude do sinal através da equação (114). Essa equação é implementada no modelo pelos blocos de fundo verde, e para o cálculo do ângulo de fase através da equação (115), implementada no modelo através dos blocos de fundo vermelho. A componente CC é obtida conforme a equação (116), diretamente do valor do terceiro elemento do vetor \vec{c} .

Figura 72 - Trecho de script MATLAB para criar a matriz mA

```

26      %preenche matriz mA
27      mA = ones(buffer_size,3);
28      mA(1,1)=0;
29
30      T_amostragem = 1/frequencia_amostragem;
31      omega = 2*pi*frequencia;
32      OMEGA = omega*T_amostragem;
33
34      for i=2:buffer_size
35          n = i-1;
36          mA(i,1) = sin(OMEGA*n);
37          mA(i,2) = cos(OMEGA*n);
38      end;

```

O dimensionamento do tamanho do *buffer* foi feito seguindo o critério do algoritmo Radix-4 de se utilizar uma potência 4^N , implicando em uma frequência de amostragem como definida pela equação (136). O sinal defasado testado foi o sinal senoidal definido com os parâmetros da equação (138) e com frequência de 50 kHz. O tempo final de simulação foi estabelecido o equivalente a 256 ciclos do sinal defasado. O solver a ser utilizado é o solver de passo variável, “discrete (no continuous states)” com passo máximo de tempo configurado como automático. A simulação é executada variando-se a fase do sinal defasado no intervalo $[0^\circ, 360^\circ]$ com passo $0,1^\circ$. O tamanho do buffer escolhido inicialmente foi de 4^4 , o que equivale a 256 pontos. A estimativa de fase obtida pelo modelo é mostrada na Figura 73 e a estimativa de magnitude obtida pelo modelo é mostrada na Figura 74.

Figura 73 - Fase estimada pelo modelo de ajuste de senóide para 256 pontos de amostragem

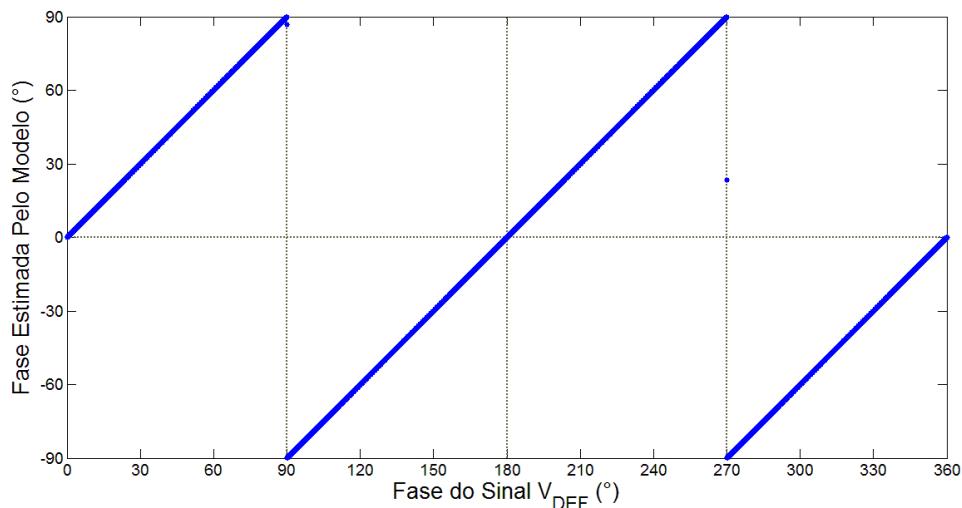
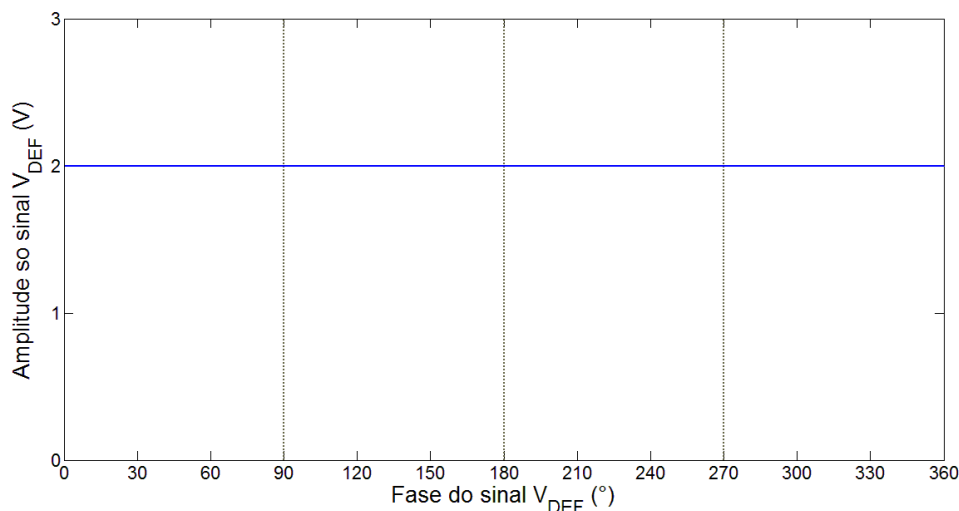


Figura 74 - Amplitude do sinal defasado estimado pelo modelo de ajuste da senóide para 256 pontos de amostragem



A curva de fase estimada pelo modelo de ajuste de senóide está de acordo com a resposta da função arco-tangente de dois quadrantes mostrada na Figura 28. O valor de magnitude obtida não apresenta erros e está de acordo com o valor de amplitude utilizado na equação (138).

3.11 SELEÇÃO DAS TÉCNICAS DE MEDIÇÃO DE FASE PARA IMPLEMENTAÇÃO EM VHDL

3.11.1 CRITÉRIOS DE SELEÇÃO

O primeiro critério levado em consideração para seleção da técnica a ser implementada é a complexidade matemática. Cálculos de multiplicação de números grandes e/ou muito pequenos, divisão e multiplicação fracionária, radiciação e exponenciação, necessitam de unidades de aritmética de ponto flutuante. Operações como logaritmos, senos, cossenos, tangentes, bem como suas operações inversas (arco-seno, arco-cosseno e arco-tangente), necessitam de tabelas de valores armazenadas em ROM com algoritmo de interpolação à base de aritmética de ponto flutuante.

Existem *ip cores* disponíveis para uso em conjunto com o código VHDL, porém, estes núcleos demandam um número significativo de recursos e de pulsos de *clock* para terminar a execução de uma operação completa. A exemplo, foi testado o *ip core* de código aberto para matemática de ponto flutuante “FPU100 Version 19” disponibilizado por Al-Eryani (2009), o qual realiza as quatro operações básicas e o cálculo de raiz quadrada. Para esse *ip core*, obteve-se o uso de recursos mostrado na Tabela 4.

Tabela 4 - Consumo de recursos do FPGA pelo *ip core* FPU100 Versão 19

Utilização Lógica	Usado	Disponível	Utilização
Número de Slice Registers	1576	28800	5%
Número de Slice LUTs	7161	28800	24%
Número de fully used LUT-FF pairs	894	7843	11%
Número de bonded IOBs	112	480	23%
Número de BUFG/BUFGCTRLs	1	32	3%

O segundo critério levado em consideração para a seleção da técnica é quais são as frequências de *clock* envolvidas na operação da topologia. Certas configurações necessitam de frequências de *clock* auxiliares que não podem ser obtidas simplesmente através da divisão de frequência baseada na contagem dos pulsos de *clock* do FPGA. Nesse caso, é necessária a utilizações de geradores de *clock* externos ao núcleo central do FPGA, os quais são programados através de *ip cores*. Ainda, frequências elevadas aumentam o consumo de potência tornando-se um elemento nocivo no desenvolvimento de aplicações de baixo consumo.

A relação sinal ruído também é significativa para a definição da topologia. Se a relação sinal ruído for alta, existe a necessidade de topologia adicional para tratamento dos sinais de entrada, de modo a reduzir significativamente o efeito do ruído. Métodos que trabalham com nuvens de pontos conseguem ter imunidade significativa a ruídos. Já os métodos de medição direta são altamente sensíveis a qualquer perturbação, demandando um bom pré-condicionamento de sinal.

Do ponto de vista de requisitos de a técnica selecionada deve permitir que os requisitos de desempenho sejam cumpridos. Desta forma, deve-se verificar se a técnica selecionada é capaz de fornecer a resolução de medida desejada, assim como se existem possíveis limitações relacionadas aos intervalos de medida causando descontinuidades na faixa de valores de interesse.

3.11.2 REQUISITOS A SEREM CUMPRIDOS PELAS TOPOLOGIA SELECIONADAS

O foco da seleção da topologia é a sua utilização pra a medição de fase e de amplitude de um sinal medido, obtido através da técnica de bioimpedância de frequência única (BIA), na frequência padrão de 50 kHz, cujos valores serão utilizados para a determinação da bioimpedância de um tecido biológico.

As topologias a serem implementadas em FPGA deverão executar a função dos blocos detectores de amplitude e dos blocos detectores de fase do sistema de medição de bioimpedância genérico mostrado na Figura 4. Dessa forma, as topologias selecionadas devem ser capazes de fornecer um valor representativo da fase e um valor representativo da amplitude do sinal medido.

Supõe-se que os valores dos sinais medidos são obtidos através de ADCs, os quais possuem uma resolução de discretização suficiente para permitir a representação das tensões obtidas na faixa de resistividade típica de 200 Ω a 800 Ω (conforme abordado no capítulo 1), com a passagem de uma corrente elétrica alternada de 1 mA.

A topologia selecionada para a determinação do valor de magnitude do sinal medido, deve ter resolução numérica suficiente para permitir a representação numérica de valores de tensão entre -0,8 V a +0,8 V. Esses valores foram obtidos para uma corrente alternada de magnitude 1 mA aplicada à faixa de resistividade típica, considerando que o ADC recebe o sinal medido com ganho unitário.

Da mesma forma, a topologia selecionada também deve permitir a representação numérica para a medição da reatância capacitiva, entre o intervalo angular de $[-90^\circ, 90^\circ]$. A resolução mínima de representação deve ser

de $0,1^\circ$, podendo ser ampliada até $0,01^\circ$ caso o impacto na utilização de recursos e no consumo de potência não seja significativo.

3.11.3 COMPARAÇÃO ENTRE AS TÉCNICAS

Para se efetuar a comparação das técnicas, o processo lógico-aritmético de determinação de fase e de amplitude do sinal, foi analisado e descrito conforme os parâmetros de caracterização mostrados na Tabela 5.

Dentro dos parâmetros são incluídos ainda a resolução angular obtida durante a simulação em Simulink, imunidade de ruído teórica (cuja obtenção é mostrada no apêndice A), a necessidade do uso de sinal de referência para execução da medição e se o sinal de referência pode ser uma forma de onda quadrada. Os espaços não preenchidos na tabela indicam que ou o recurso não é utilizado, ou que o recurso não é admissível.

Dada a natureza das informações apresentadas na Tabela 5, a mesma se torna bastante útil para a seleção de método a ser implementado. Através dela obtém-se os dados necessários para avaliação da eficiência computacional de cada método. A quantização das operações aritméticas apresentada na tabela foi feita considerando-se a obtenção dos sinais representativos de magnitude e fase e não seus valores finais calculados representando o ângulo e a amplitude.

A Tabela 6 mostra o número de operações aritméticas necessárias para operações matemáticas mais complexas, tais como: cálculo de determinantes, cálculo de matriz inversa, resolução de sistemas lineares, determinação de autovalores e auto vetores, e operações com números complexos na forma cartesiana. Os valores foram quantizados com base na solução analítica de cada operação matemática complexa, cujo número de operações matemáticas básicas é menor.

Tabela 5 - Parametrização do processo lógico-aritmético das técnicas de medição de magnitude e fase simuladas em Simulink

Id	Técnica								
1	Detector de Magnitude								
2	Figura de Lissajous								
3	Ajuste de Elipse								
4	Demodulação de Quadratura – FIR Serial								
5	Medição Direta Digital por Pulsos								
6	PWM – Modulação de Largura de Pulso								
7	DFT - Transformada Discreta de Fourier Radix4								
8	Ajuste de de Senóide								
Id	Parâmetro								
1	Imunidade a ruído (dB)	56	56	43	49	42	30	55	-
2	Necessita sinal de referência		Sim	Sim	Sim	Sim	Sim	Sim	
3	Admite sinal de referência quadrado			Sim	Sim			Sim	
4	Número de sinais de referência		1	1	1	2	1	1	
5	Admite <i>Scaling</i>					Sim			Sim
6	Frequência de amostragem necessária	$4^4 f$	$4^5 f$			$40 \cdot f$	$100 \cdot f$	$3600 \cdot f$	$100 \cdot f$
7	Frequência de <i>clock</i> auxiliar				$3600 \cdot f$				
8	Mínimo armazenamento de amostras	256	1024			80	100	2	1
9	Número de variáveis em RAM	256	5120			80	800		
10	Número de variáveis em LATCH	6	12		1	4	102	6	3
11	Número de coeficientes (ROM)	786	2048			80			
12	Operações de Soma	65286	11776		≤ 3600	80	88347	3600	100
13	Operações de Subtração		14848				54		
14	Operações de Multiplicação	65545	28672			80	90614		
15	Operações de Divisão						18	7200	100
16	Operações de Radiciação						4		
17	Grau máximo de radiciação						3		
18	Cosseno						3		
19	Arco-cosseno						1		
20	Resolução angular obtida	0.1°	0.1°	0.57°	0.1°	0.1°	0.1°	0.1°	

Tabela 6 - Quantização das operações aritméticas básicas para operações matemáticas compostas

Operação	Multipli- cações	Adições	Subtra- ções	Divisões	\sqrt{x}	$\sqrt[3]{x}$	$\cos^{-1} \emptyset$	$\cos \emptyset$
$M_{n \times 3} \cdot M_{3 \times n}^T$	$3 \cdot n^2$	$3 \cdot (n - 1) \cdot n$						
$ M_{3 \times 3} $	12	4	1					
$inv(M_{3 \times 3})$	30	4	10	1				
$p(x) = 0$ $deg = 3^\circ$	4	3	2	2	2	2	1	3
Sistema Linear $A_{3 \times 3} \cdot \vec{x} = \vec{B}$	48	16	4	3				
$eigval(M_{3 \times 3})$	20	14	2	2	2	2	1	3
$eigvec(M_{3 \times 3}, \lambda_i)$	48	16	7	3				
$\vec{a}_n \cdot \vec{b}_n$	n							
$M_{n \times 3} \cdot \vec{b}_n$	n^2	$(n - 1) \cdot n$						
$A_{m \times n} \cdot B_{m \times n}$	$m \cdot n^2$	$m \cdot (n - 1) \cdot n$						
$A_{m \times n} + B_{m \times n}$		$m \cdot n$						
$A_{m \times n} - B_{m \times n}$			$m \cdot n$					
$c \cdot A_{m \times n}$	$m \cdot n$							
$\frac{1}{c} \cdot A_{m \times n}$				$m \cdot n$				
Forma cartesiana $Z_1 + Z_2$		2						
Forma cartesiana $Z_1 - Z_2$			2					
Forma cartesiana $Z_1 \cdot Z_2$	4	1	1					
Forma cartesiana Z_1 / Z_2	6	3		2				

Considerando o *ip core* FPU100 versão 19 como o responsável pelas quatro operações aritméticas básicas mais radiciação, obtém-se a Tabela 7 a qual mostra a quantidade de pulsos de *clock* necessários para cada tipo de operação aritmética do *ip core*.

Tabela 7 - Pulsos de *clock* necessários para cada operação aritmética do *ip core* FPU100 versão 19

Operação	Número de Pulsos de <i>Clock</i>
Adição	9
Subtração	9
Multiplicação	14
Divisão	36
Radiação com grau 2	36

Considerando que o valor de cosseno e arco-cosseno são calculados com algoritmo CORDIC (algoritmo teórico não sendo de bibliotecas prontas) utilizando 10 iterações, tem-se o uso final de operações aritméticas básicas mostradas na Tabela 8 para o cálculo de cosseno e arco-cosseno.

Tabela 8 - Quantidade de operações aritméticas básicas necessárias para o algoritmo CORDIC com 10 iterações

Operação	Multiplicações	Adições	Subtrações	Divisões
Cosseno	61	20	10	40
Arco-Cosseno	61	20	10	40

Com base nos valores de operações aritméticas utilizadas em cada técnica e com base nos pulsos de *clock* utilizados pelo *ip core* FPU100 versão 19, obtém-se a Tabela 9. Ela mostra para cada técnica, uma aproximação da quantidade de pulsos de *clock* necessários para se completar toda a operação de estimativa de fase e de amplitude (quando aplicável). Essa aproximação foi calculada considerando-se que não existe a utilização de *pipelining* e apenas um *ip core*.

Tabela 9 - Total de pulsos de *clock* utilizados por cada técnica de medição

Método	Total de Pulsos de <i>Clock</i>
Lissajous	291.600
Ajuste de elipse	2.075.613
Demodulação de quadratura	1.840
PWM	0
DDM – Contagem de pulsos	32.400
DFT – Radix-4	641.024
Ajuste de senóide	1.505.204
Detector de Magnitude	4.500

Dentre os métodos que trabalham com nuvens de ponto, o método da demodulação de quadratura é o que apresentou a menor utilização de pulsos de *clock*. Portanto, é mais eficiente computacionalmente. O número de amostras necessárias é menor quando comparado aos outros métodos de nuvens de pontos, apresentando também menor consumo de ROM e menor consumo de RAM. Os sinais estimados por esse método são: a diferença de fase em radianos e metade da amplitude do sinal defasado.

O método do ajuste da elipse é computacionalmente o pior método de nuvens de ponto, apresentando o maior uso de pulsos de *clock* para execução completa da determinação dos coeficientes da elipse. Sua saída consiste nos coeficientes da elipse. Também, esse método falha em representar ângulos de fase próximos de 0° e $\pm 180^\circ$. Deste modo, verifica-se que esse método não é o mais indicado para a implementação desejada, pois busca-se um método de baixo-consumo.

O método do ajuste da senóide se apresentou como o segundo pior método de nuvens de ponto computacionalmente eficiente. A saída desse método são três coeficientes utilizados para se determinar a amplitude, a fase e a componente CC da senóide.

O método da transformada discreta de Fourier com algoritmo Radix-4, com 1024 pontos de amostragem, se apresentou como o segundo método de nuvens de ponto mais eficiente computacionalmente. A sua saída é um conjunto de pontos complexos no domínio da frequência. O ponto de interesse é o ponto com o maior valor em módulo. Se o algoritmo Radix-4 for alterado para guardar apenas o número calculado que se mostra maior em módulo, pode-se diminuir o uso de memória RAM em 2/3.

Dentre todos os métodos de medição direta, a técnica da modulação de largura de pulso – PWM, se mostrou a mais eficiente computacionalmente pois a mesma não efetua cálculo de nenhuma espécie, apenas gerando uma forma de onda a qual é aplicada à um filtro externo. Os filtros utilizados na simulação em Simulink para o modelo desta técnica não resultaram na resolução angular de requisito ($0,1^\circ$). A implementação do filtro de forma digital ou a utilização de um ADC com um bloco para calcular o valor médio do sinal medido pode melhorar o resultado de forma a se obter um valor de resolução angular próximo ou melhor que a resolução angular de requisito.

Do ponto de vista da eficiência computacional, o método derivado da técnica de Lissajous, se mostrou como o pior dentre os métodos de medição direta. Esse método é implementado em conjunto com o detector de magnitude (o qual também está parametrizado na Tabela 4). A sua frequência de amostragem deve ser definida de modo a permitir a discretização do período do sinal medido na resolução angular desejada, conforme as equações (75) e (76). As saídas de medição desse método são a amplitude do sinal defasado, e uma razão que representa o seno da fase do sinal defasado.

O segundo método mais eficiente computacionalmente dentre os métodos de medição direta é o método da contagem de pulsos. Esse método fornece o valor da fase do sinal defasado em pulsos contados. Esse valor deve ser multiplicado pelo valor da resolução angular para se obter o ângulo em graus. Para resoluções angulares na potência de dez, basta inserir-se um separador decimal entre os algarismos (antes e depois da casa decimal) diretamente na exibição do resultado sem a necessidade de cálculos de conversão.

O método do detector de magnitude serve apenas para determinar qual a amplitude máxima do sinal defasado, sendo sua saída, diretamente o valor da amplitude do sinal medido. Esse método normalmente é utilizado de forma conjugada com o método de PWM, com o método da técnica de Lissajous e com o método da contagem por pulsos.

Uma vez que os métodos simulados podem ser divididos em duas categorias: os que trabalham com nuvens de ponto e os que trabalham com medição direta; com base na análise de eficiência computacional, foram escolhidos um método de cada categoria para implementação em VHDL. O método de nuvens de ponto escolhido foi o método da demodulação da quadratura pois é o método mais eficiente computacionalmente e com menor uso de recursos. O método de medição direta selecionado foi o método da contagem de pulsos, pois apesar de ser o segundo método de medição direta mais eficiente computacionalmente, ao contrário do PWM, permite comparação direta. Devido à escolha do método da contagem de pulsos, faz-se necessária a implementação do detector de magnitude de forma conjugada ao detector de fase.

4. VERIFICAÇÃO DO CÓDIGO VHDL E VALIDAÇÃO EM *HARDWARE*

Este capítulo introduz de forma breve a ferramenta System Generator e o seu conceito de co-simulação (execução do código VHDL sintetizado dentro do FPGA, porém utilizando emulação em *software* para gerar os estímulos introduzidos nos pinos de entrada do FPGA, e para processar os sinais lidos nos pinos de saída), ambos utilizados para o teste do código VHDL. Em seguida, são mostradas as considerações feitas para a implementação em código VHDL das topologias selecionadas no capítulo anterior, os resultados de simulação obtidos com o System Generator durante a verificação do código VHDL via emulação em *software*, e os resultados obtidos através da co-simulação em *hardware* para a validação em FPGA do código VHDL.

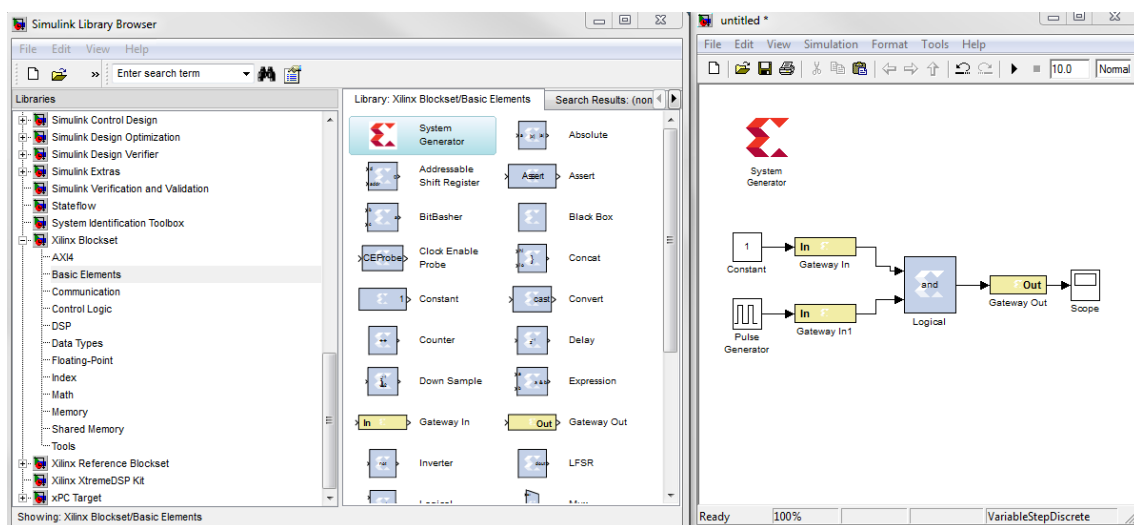
4.1 SYSTEM GENERATOR

O System Generator é uma ferramenta da XLINK na forma de uma *toolbox* para o Simulink. Esta *toolbox* é composta por uma variedade de blocos que implementam dentro de si código VHDL gerado automaticamente, e que podem ser conectados aos blocos das outras toolboxes disponíveis no Simulink. Dessa forma, obtém-se a integração de um modelo matemático Simulink ao modelo em código VHDL que se quer testar. Como vantagem, a simulação e a verificação dos resultados da topologia VHDL são feitas de forma facilitada, podendo inclusive serem automatizadas. Dentre os blocos disponibilizados pelo System Generator, existe um bloco denominado “Black Box”, o qual permite importar um código VHDL criado de forma manual ou gerado de forma automática por outro aplicativo, para dentro da topologia VHDL a ser testada.

A Figura 75 apresenta a interface básica do System Generator, exemplificando a implementação de uma porta E em FGPA. O bloco denominado “System Generator” é de uso obrigatório em qualquer modelo que seja construído com os blocos do System Generator, e é o responsável por informar ao Simulink, que parte da simulação deve ser executada pelo System Generator.

Ainda, este bloco configura qual é o modelo de placa de FPGA que será utilizada na simulação e qual a sua frequência de *clock*. Os blocos do tipo *Gateway* são utilizados para se fazer a interface entre os pinos de entrada e saída da entidade que seria especificada no código VHDL *top level* e os blocos do tipo Simulink. O bloco “Logical” implementa as funções lógicas básicas, sendo que para o exemplo apresentado, a função lógica do bloco foi configurada para uma função E. Os demais blocos são blocos de uso comum do Simulink.

Figura 75 - Interface básica do System Generator dentro do Simulink

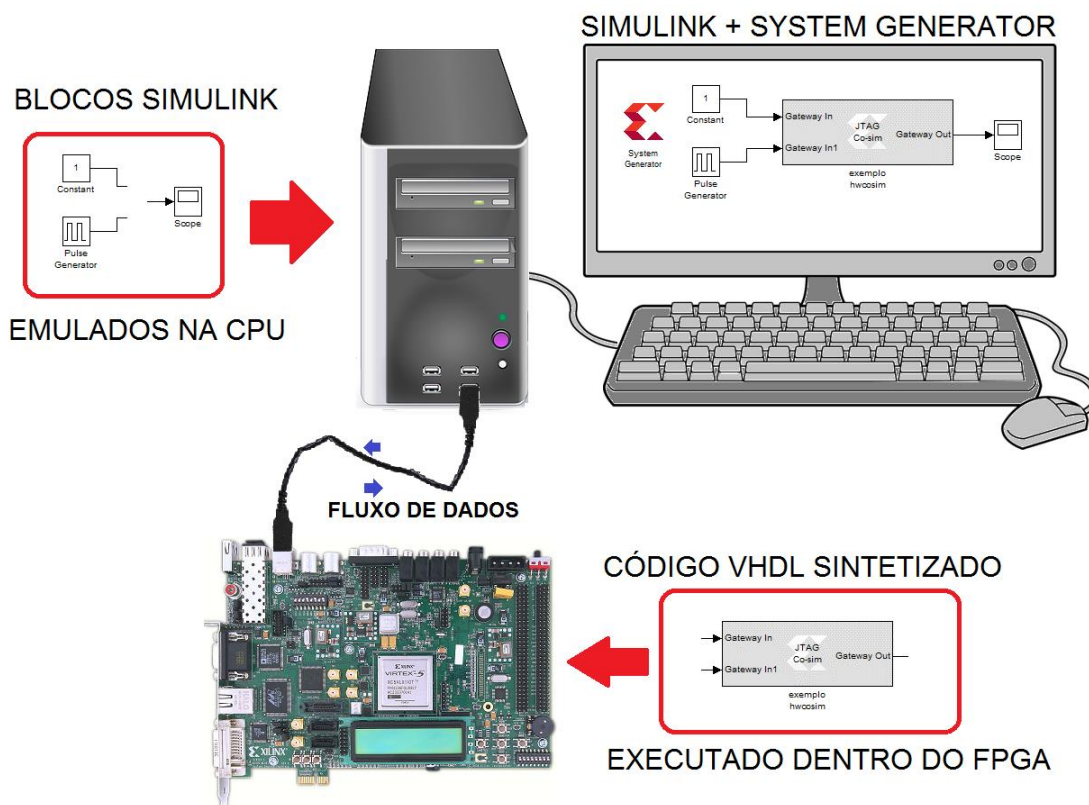


Quando se solicita ao Simulink que execute a simulação do modelo, o Simulink verifica se o modelo irá ser simulado pela primeira vez, ou se houve alguma alteração na topologia composta pelos blocos do System Generator. Caso alguma dessas duas condições tenha sido identificada, todo código VHDL é sintetizado por um aplicativo do System Generator, e em seguida, o Simulink inicia a simulação de fato, seguindo as configurações especificadas para o *solver* na simulação dos blocos do Simulink, e as configurações especificadas no bloco “System Generator” para os blocos de código VHDL. Enquanto o Simulink executa a simulação da parte matemática, o *software* do System Generator executa a simulação do código VHDL via emulação totalmente em *software*.

4.2 VALIDAÇÃO EM HARDWARE VIA CO-SIMULAÇÃO

A co-simulação de *hardware* disponibilizada pelo System Generator consiste na execução do código VHDL sintetizado dentro da placa do FPGA ao invés da emulação via *software*. A Figura 76 ilustra o conceito da co-simulação utilizado para a validação de código em *hardware*. Conforme a representação, um *desktop* rodando o Simulink com a *toolbox* System Generator gera os sinais de teste que são enviados ao FPGA. Em seguida, o FPGA processa os sinais recebidos e produz nos seus pinos de saída a resposta de acordo com o código VHDL que foi sintetizado e previamente embarcado para dentro do FPGA. Finalmente, o Simulink recebe os sinais dos pinos de saída do FPGA e executa o processamento desses sinais para determinação dos valores de interesse. O FPGA utilizado para a validação de código em *hardware* foi uma placa Virtex-5 baseada no FPGA LX50T e a troca de dados entre a placa do FPGA e o *desktop* foi feita através da porta USB da placa do FPGA.

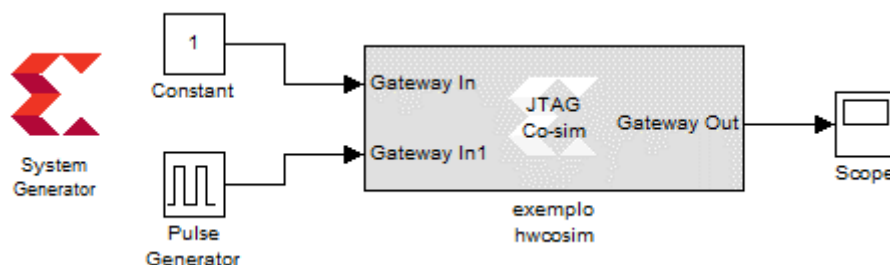
Figura 76 - Visão geral do conceito de co-simulação para validação em *hardware* do código VHDL com processamento de sinais pelo Simulink



Para que a co-simulação de *hardware* seja feita, primeiramente, o modelo Simulink que contém os blocos do System Generator deve ser compilado, gerando o arquivo de *bitstream*. A compilação é executada, pressionando-se o botão “generate” dentro do bloco “System Generator”, bloco o qual, deve obrigatoriamente estar dentro do modelo Simulink. Após a compilação, o System Generator exporta um novo bloco para dentro da biblioteca do Simulink, sendo que esse bloco, referencia o arquivo de *bitstream* do código VHDL sintetizado.

Cria-se então um novo modelo, no qual o bloco que foi exportado para a biblioteca substitui todos os blocos que eram da *toolbox* do System generator, mantendo-se os blocos das demais *toolboxes* do MATLAB. Esse novo modelo quando simulado, tem todo o código do arquivo *bitstream* executado dentro do FPGA. A Figura 77 mostra o modelo de co-simulação de *hardware* obtido para o exemplo da Figura 75.

Figura 77 - Modelo de co-simulação de hardware para a porta E apresentada na Figura 75

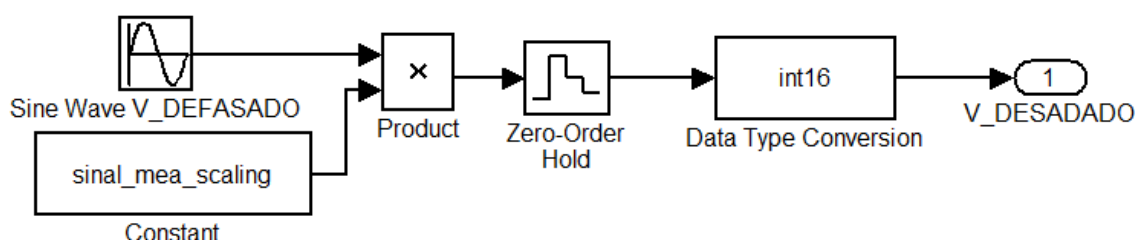


Quando se solicita o início da simulação, o arquivo de *bitstream* é embarcado automaticamente para dentro do FPGA. Como o próprio System Generator cria um arquivo de especificação de portas para controlar a troca de dados entre os blocos Simulink e o FPGA, não é necessária a criação de nenhum código VHDL adicional para execução desse procedimento.

4.3 MODELAGEM COMPORTAMENTAL DO ADC

Não haviam ADCs disponíveis na placa de FPGA utilizada, nem na forma de acessório externo para integração com a placa. Sendo assim, como alternativa, utilizou-se da modelagem comportamental mostrada na Figura 78, para se criar virtualmente um gerador de sinal conjugado com um ADC.

Figura 78 - Topologia do ADC virtual utilizado para simulação.



4.4 DEMODULADOR DE QUADRATURA

No demodulador de quadratura (DQ), para a obtenção de 40 amostras do sinal de 50 kHz, a frequência de amostragem deve ser de 2 MHz, o que equivale a um tempo de amostragem de 0,5 microssegundos. A sua topologia é composta de dois ramos paralelos, sendo que cada ramo consiste de um misturador em série com um filtro passa-baixa. Se o filtro passa-baixa utilizado é um filtro FIR serial de ordem 40 e a aritmética de ponto flutuante é realizada por um *ip core* FPU100 distinto para cada ramo, o número de pulsos de *clock* gastos em cada ramo para completar toda a operação de cálculo é equivalente à metade do valor contabilizado na Tabela 9.

A partir da Tabela 9 obtém-se o valor de 920 pulsos de *clock* para que toda a operação de cálculo por ramo seja concluída. A frequência padrão do FPGA é de 100 MHz, logo, os 920 pulsos de *clock* necessários para se completar toda a operação do misturador + filtro gastam o tempo total de 9,2 microssegundos, implicando que o tempo necessário para se completar o cálculo é maior que o tempo no qual se obtém uma nova amostra.

Como alternativas para solucionar o fato de que o tempo total de execução de cálculo é maior que o tempo de amostragem, tem-se: aumentar o *clock* do FPGA, a utilização de um *clock* auxiliar mais rápido para a FPU, a utilização de uma topologia de filtro FIR totalmente paralela ou de uma topologia parcialmente serial, e o uso de *scaling*.

A utilização de um *clock* maior para o FPGA ou a utilização de um *clock* auxiliar mais rápido para a FPU vão acarretar em maior consumo de potência. A utilização de uma topologia totalmente paralela para o filtro FIR é impraticável pois seriam necessários 160 *ip cores* FPU100 os quais utilizariam 1.145.760 *slices* de LUT enquanto que o FPGA utilizado disponibiliza apenas 28.800 *slices*.

Uma topologia parcialmente serial necessitaria de uma maior quantidade de *ip cores* FPU100, sendo que a menor topologia parcialmente serial resultaria em dois *ip cores* FPU100 por ramo. Nesse caso, o demodulador de quadratura necessitaria de quatro *ip cores* FPU100, resultando no uso de 28.644 *slices* LUT, ou seja, consumiria praticamente todos os recursos disponíveis no FPGA, causando grande ocupação de área.

A opção de *scaling* consiste na conversão dos valores dos coeficientes do filtro FIR e dos valores numéricos de tensão nas entradas do misturador em números inteiros. Os números são obtidos através da multiplicação dos valores originais em ponto flutuante por uma potência de dez que desloque a quantidade de casas decimais desejadas para à esquerda do ponto decimal.

A utilização de *scaling* é a solução mais vantajosa pois o valor de saída do ADC normalmente é um valor numérico inteiro, as operações aritméticas de números inteiros são executadas dentro do DSP do FPGA dentro do tempo de um único pulso de *clock* do FPGA. Além disso, o acréscimo de área devido ao DSP não é tão grande quanto ao que seria obtido com a utilização de vários *ip cores* FPU100. Desse modo, para o demodulador de quadratura, utilizou-se a técnica de *scaling* para a implementação da topologia em VHDL.

4.4.1 DIMENSIONAMENTO DO SCALING

De acordo com os requisitos de projeto, a entrada do misturador deve ser capaz de receber valores numéricos representativos da faixa de tensão entre $-0,8\text{ V}$ a $+0,8\text{ V}$. Considerando-se a resolução de 1 mV , um fator de *scaling* de 1000 representaria essa faixa de tensão como sendo uma faixa numérica de -800 a $+800$. Para que a amplitude na saída do filtro FIR corresponda a metade da amplitude do sinal medido, o sinal de referência deve ter amplitude de 1 V . Logo, a entrada do sinal de referência do misturador deve variar na faixa de -1 V a $+1\text{ V}$. Para o fator de *scaling* igual a 1000, tem-se a faixa numérica de -1000 a $+1000$.

Para se representar valores de -800 a $+800$ e -1000 a $+1000$ com representação em complemento de dois, são necessários pelo menos 11 *bits*, resultando na faixa de valores numéricos de -1024 a $+1023$.

O misturador funciona executando a multiplicação entre o sinal de referência e o sinal medido. Logo, os valores mínimo e máximo na saída do misturador são $-800,000$ e $+800,000$ respectivamente. Utilizando-se a representação de números inteiros com complemento de dois, são necessários pelo menos 21 *bits* para poder se representar essa faixa numérica, obtendo-se uma faixa de valores de $-1.048.576$ e $+1.048.575$. Porém, como os números que são multiplicados pelo misturador, possuem ambos 11 *bits*, o número resultante da multiplicação terá 22 *bits*, obtendo-se a faixa de valores $-2.097.152$ a $+2.097.151$.

O projeto do filtro FIR é executado utilizando-se a ferramenta *FDATool* do Simulink a qual calcula os coeficientes do filtro e exporta o código VHDL não sintetizável. Para o filtro passa baixa FIR de ordem 40, projetado com o método da janela, utilizando-se uma janela Hann para escalonar a banda passante, frequência de corte de 1 kHz e frequência de amostragem de 2 MHz , são obtidos os coeficientes de filtro mostrados na Tabela 10. A tabela mostra também os coeficientes obtidos com um escalonamento de 10^7 . O maior valor de coeficiente escalonado é de 500.043, o qual necessita de 20 *bits* para ser representado em complemento de dois.

Tabela 10 - Coeficientes do filtro FIR de ordem 40

Coeficientes FDATool	Coeficientes Escalonados
0,000000000000000000	0
0,000307635196747000	3.076
0,001223040239336050	12.230
0,002723775936177390	27.238
0,004772975212933770	47.730
0,007320250137897390	73.203
0,010302932393367900	103.029
0,013647616638379400	136.476
0,017271968733473800	172.720
0,021086754261126200	210.868
0,024998037337324100	249.980
0,028909495504996200	289.095
0,032724793632361200	327.248
0,036349958279694900	363.500
0,039695693982622400	396.957
0,042679584329302900	426.796
0,045228123547601700	452.281
0,047278528496545400	472.785
0,048780286370995000	487.803
0,049696399945941700	496.964
0,050004299646349400	500.043
0,049696399945941700	496.964
0,048780286370995000	487.803
0,047278528496545400	472.785
0,045228123547601700	452.281
0,042679584329302900	426.796
0,039695693982622400	396.957
0,036349958279694900	363.500
0,032724793632361200	327.248
0,028909495504996200	289.095
0,024998037337324100	249.980
0,021086754261126200	210.868
0,017271968733473800	172.720
0,013647616638379400	136.476
0,010302932393367900	103.029
0,007320250137897390	73.203
0,004772975212933770	47.730
0,002723775936177390	27.238
0,001223040239336050	12.230
0,000307635196747000	3.076
0,000000000000000000	0
MAIOR VALOR	500.043

A multiplicação do coeficiente do filtro pelo valor vindo da saída do misturador, resultará então em um número com 42 *bits*. Levando em conta quarenta operações de soma, se todos os números fossem o maior valor

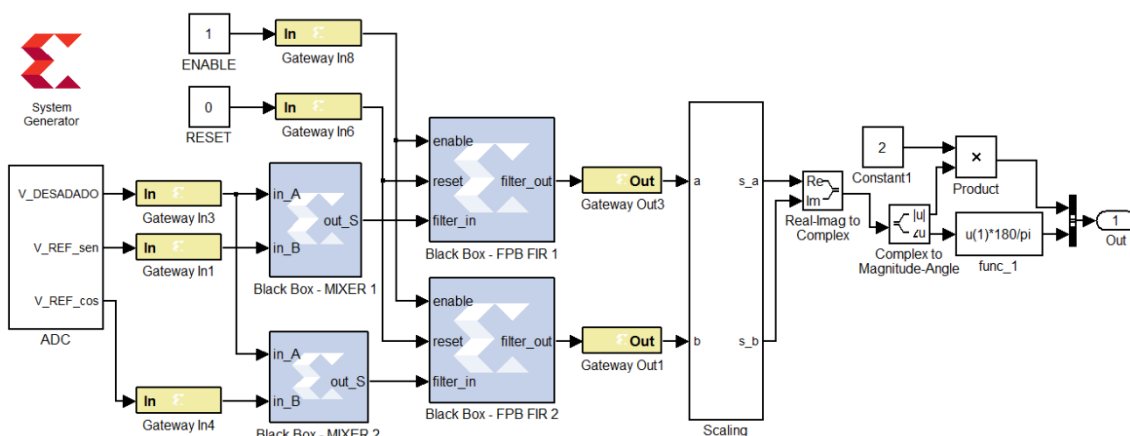
positivo, o valor resultante da soma necessitaria de 44 *bits* de representação para evitar que ocorresse *overflow*.

Buscou-se manter uma margem de representação numérica flexível o suficiente para permitir variação de parâmetros de *scaling* e prevenção de *overflow*. Assim, optou-se pela utilização de um número em complemento de dois de 16 *bits* para representar os valores na entrada do misturador e um número de 32 *bits* em complemento de dois para representar o valor dos coeficientes do filtro. Como resultado, os valores finais têm complemento de dois com 64 *bits*.

4.4.2 VERIFICAÇÃO DO CÓDIGO VHDL

O código VHDL completo do demodulador de quadratura foi feito de forma que tanto o misturador quanto o filtro são descritos em arquivos de código separados. Dessa forma, ambos têm que ser referenciados como componentes dentro de um código *top level*, o qual representa o demodulador de quadratura completo. Porém, dentro do Simulink / System Generator, tanto o misturador quanto o filtro podem ser adicionados individualmente sem a necessidade de um arquivo *top level*, resultando no modelo mostrado no modelo System Generator apresentado na Figura 79. Os blocos do tipo *Black Box* contém o código VHDL do misturador e do filtro.

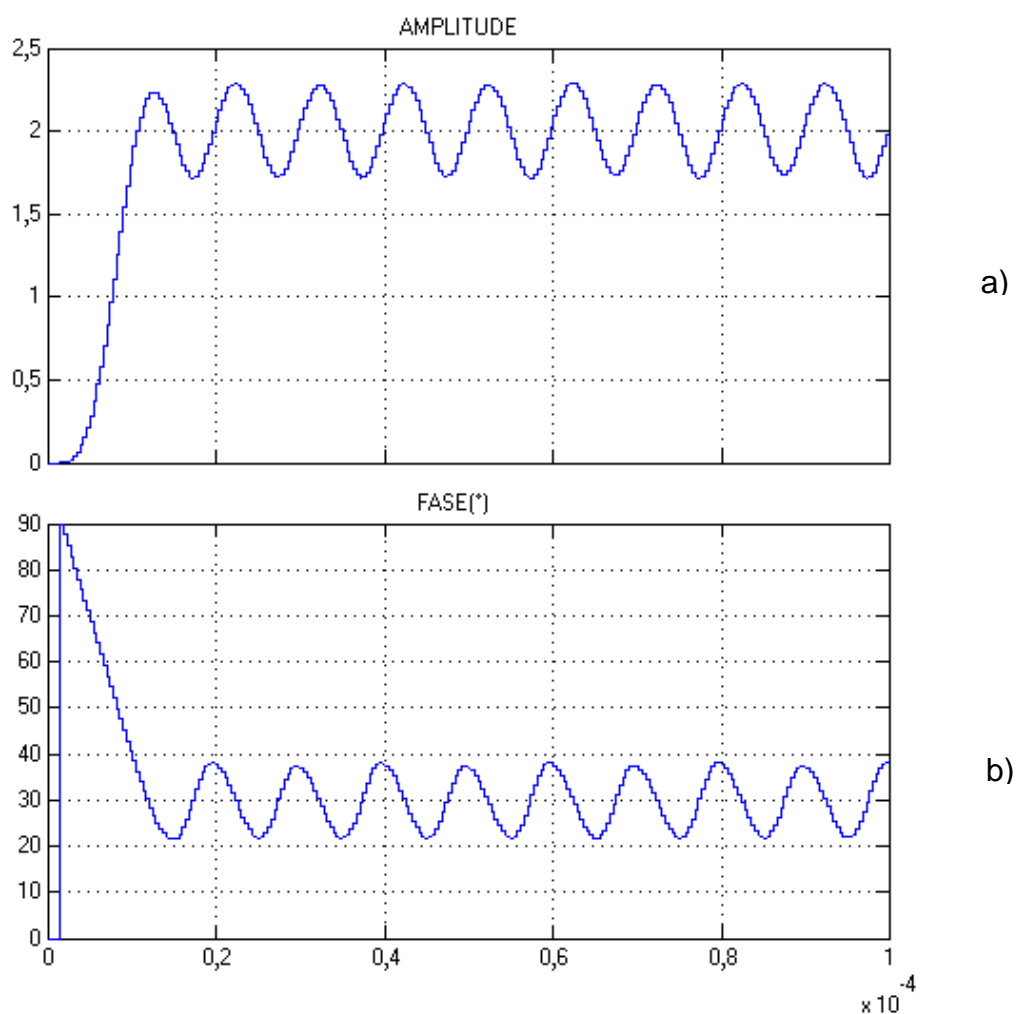
Figura 79 - Modelo System Generator do demodulador de quadratura



O modelo System Generator foi testado inicialmente para o caso padrão descrito pela equação (126), o qual é um sinal com fase 30° e amplitude 2 volts. A resposta obtida pelo modelo para a fase e para a amplitude são mostradas na Figura 80 a e b respectivamente, para um tempo total de simulação equivalente a cinco ciclos do sinal medido.

Observa-se na resposta do modelo, uma oscilação significativa de uma harmônica cujo valor é metade da frequência do sinal medido. A amplitude máxima dessa oscilação não é constante, e oscila ao redor de uma componente CC a qual se aproxima do valor correto esperado na saída do modelo. Essa oscilação é causada exclusivamente pelo filtro FIR implementado em VHDL com o uso de *scaling*.

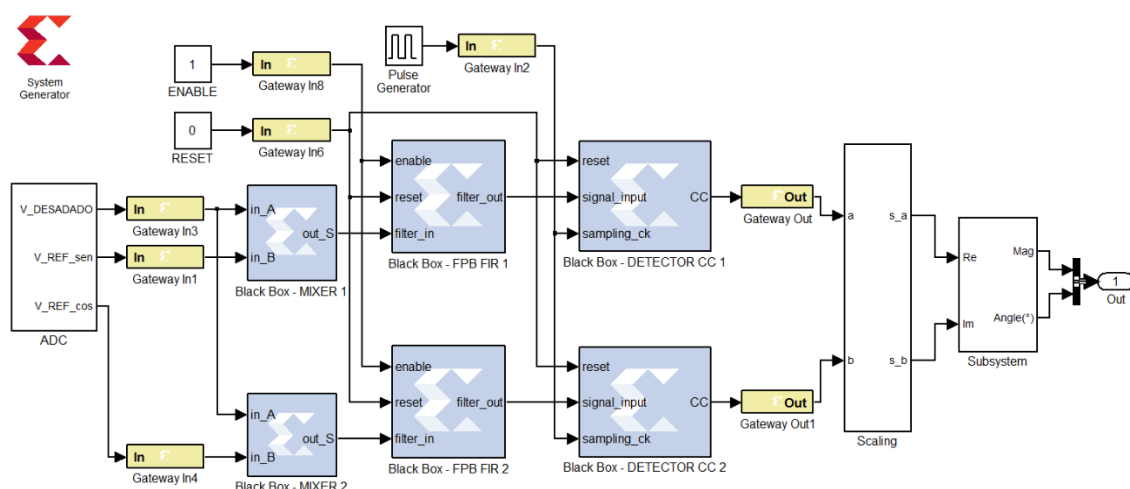
Figura 80. Resposta do modelo System Generator para o caso de teste padrão:
a) Resposta de amplitude; b) Resposta de fase.



Dado ao valor alto da amplitude da harmônica que oscila na saída do filtro, testou-se o acréscimo de um detector de componente CC para se estimar um valor médio na saída do demodulador de quadratura, a ser implementado também em código VHDL. A adaptação do detector de magnitude segundo a topologia mostrada na Figura 31 não funcionou adequadamente para o uso em 64 *bits*. A aritmética de números inteiros da biblioteca padrão do VHDL permite apenas números inteiros de até 31 *bits*. Como resultado, a não utilização de sinais do tipo inteiro, apenas vetores de bits do tipo *signed*, não permitiu a detecção correta do valor da componente CC nem do valor de vale mínimo do sinal na saída do filtro. Para ambos, foram retornados valores zero.

Como solução alternativa, foi criado um novo detector de magnitude que se baseia na comparação entre o sentido de crescimento do valor de uma nova amostra em relação ao sentido de crescimento da amostra anterior. Quando ocorre mudança de aumento de valor para diminuição de valor, é detectado o valor de um pico. Quando ocorre mudança de diminuição de valor para aumento de valor, é detectado um vale. O valor da componente CC é a média entre esses dois valores. A Figura 81 mostra o detalhe da conexão dos blocos detectores de componente CC na saída dos filtros, para o modelo apresentado na Figura 79.

Figura 81 - Inserção do detector de componente CC no modelo do demodulador de quadratura



Os sinais obtidos na saída dos filtros e na saída dos detectores de componente CC, para a simulação do caso padrão, são mostrados na Figura 82 a e b para o sinal representativo da magnitude do sinal medido, e na Figura 83 a e b para o sinal representativo de fase. Verifica-se que o efeito da inserção dos detectores de componente CC limita a variação do sinal na saída do demodulador, para o caso teste, em um valor de aproximadamente $\pm 0,075 \times 10^{12}$ tanto para a amplitude quanto para a fase. Esse valor de variação equivale a aproximadamente 6% da variação observada na saída dos filtros.

Figura 82 - Sinal representativo de magnitude: a) Sinal na saída do filtro FIR; b) Sinal na saída do detector CC.

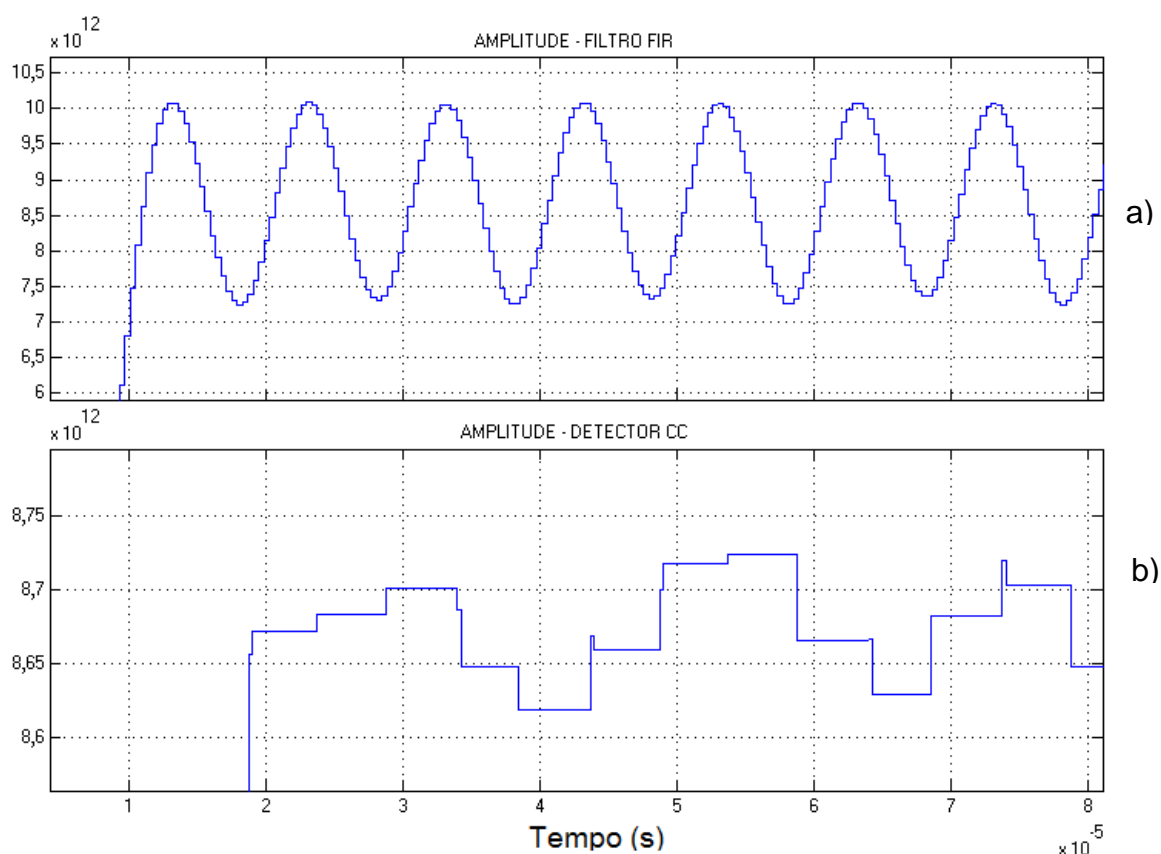
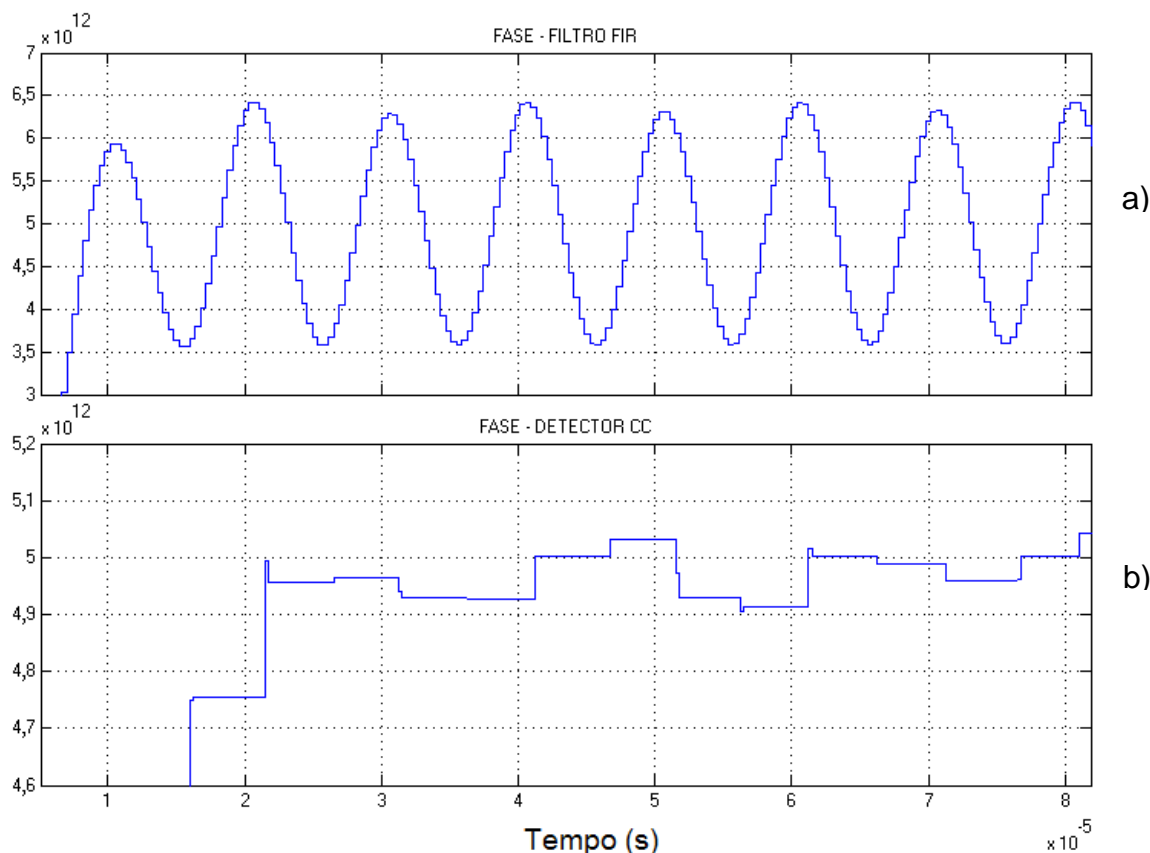


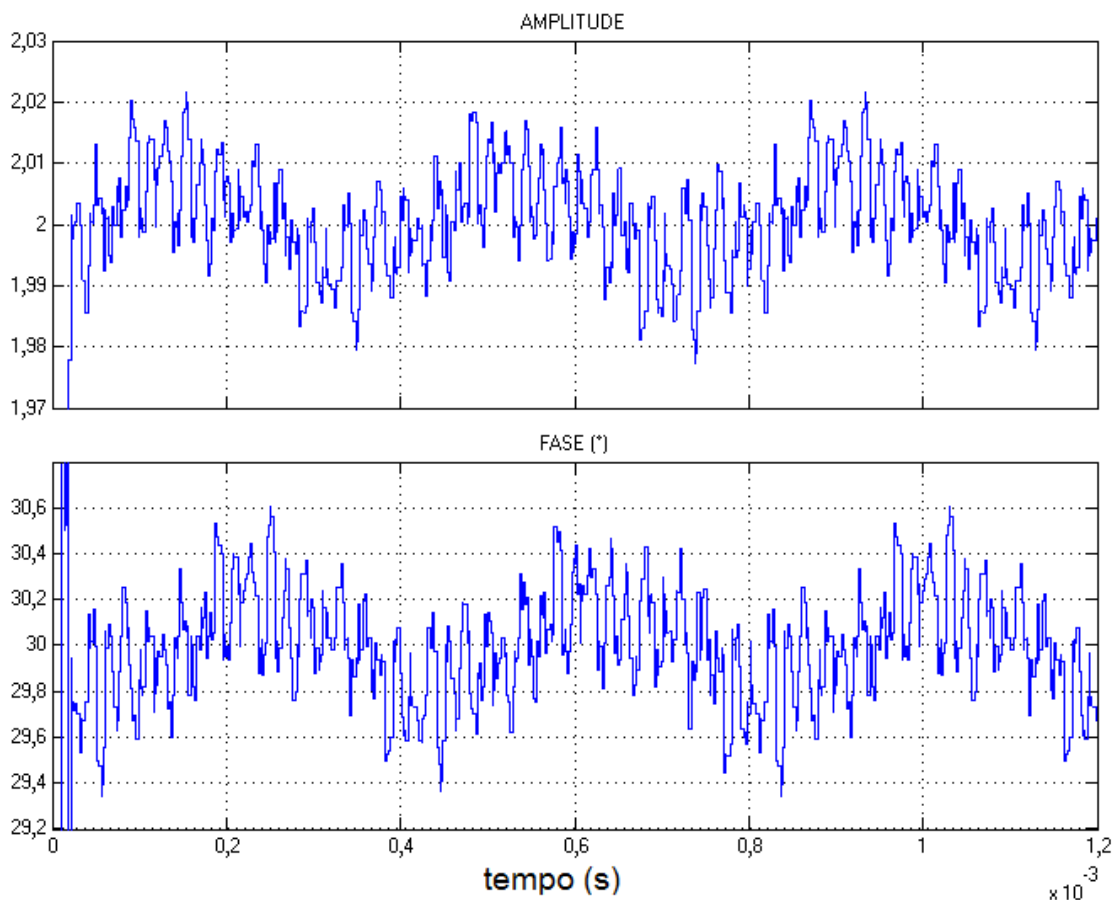
Figura 83 - Sinal representativo de fase: a) Sinal na saída do filtro FIR; b) Sinal na saída do detector CC.



Ainda para a simulação do caso padrão, a resposta da saída do modelo da topologia com a inserção dos detectores de componente CC é mostrada na Figura 84. Observa-se que a redução na variação dos sinais na saída dos filtros, obtida com a inserção dos detectores de componente CC, resultou também na redução da variação total do ângulo de fase medido na saída do modelo.

Para um tempo total de simulação equivalente a sessenta ciclos do sinal defasado, e utilizando-se os últimos cinquenta e sete ciclos para a análise de resposta do modelo, obteve-se um valor médio de fase próximo de $30,0026^\circ$, um máximo valor de fase próximo de $30,6064^\circ$ e um mínimo valor de fase próximo de $29,3459^\circ$. Esses valores equivalem a uma variação de $\pm 0,65^\circ$ em torno do valor médio de fase obtido. A topologia sem os detectores de componente CC havia resultado originalmente em um valor de variação de fase próximo de $\pm 8^\circ$.

Figura 84 - Resposta do modelo VHDL do demodulador de quadratura após a inserção dos detectores de componente CC



Devido a aparente melhora de resposta obtida com o uso de detectores de componente CC, a topologia do demodulador de quadratura com a utilização de detectores de componente CC foi simulada para um intervalo mais amplo de defasagem do sinal defasado. O intervalo escolhido foi $[0^\circ, 360^\circ]$.

Para a simulação, utilizou-se um tempo total de simulação equivalente a 20 ciclos do sinal defasado e passo de 1° . Considerando que a convergência do modelo ocorre após 3 ciclos completos do sinal defasado, estipulou-se para a simulação, que a resposta do modelo é o valor médio obtido através da média de todos os valores estimados pelo modelo durante os últimos 17 ciclos do sinal defasado. A curva de resposta de fase obtida é mostrada na Figura 85 e a curva de resposta de magnitude obtida é mostrada na Figura 86.

Figura 85 - Resposta de fase do modelo do demodulador de quadratura com detector de componente CC para o intervalo $[0^\circ, 360^\circ]$ com passo de 1°

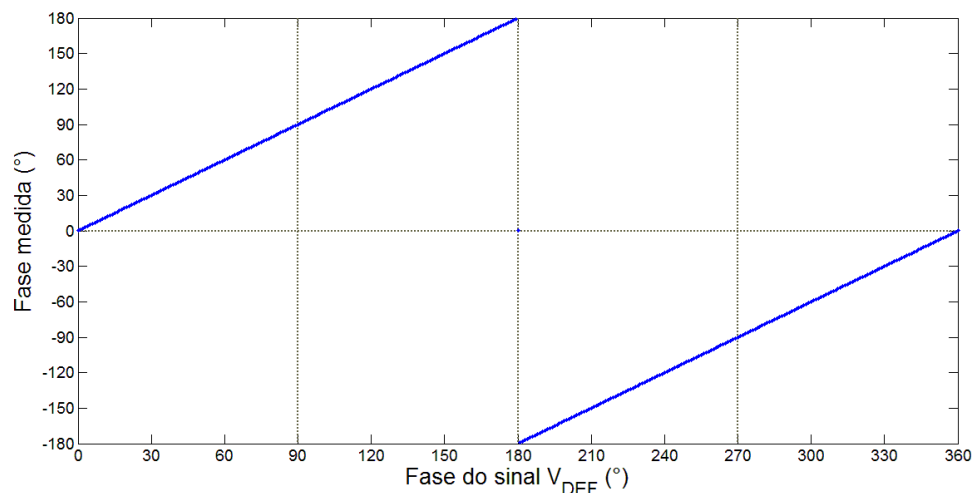
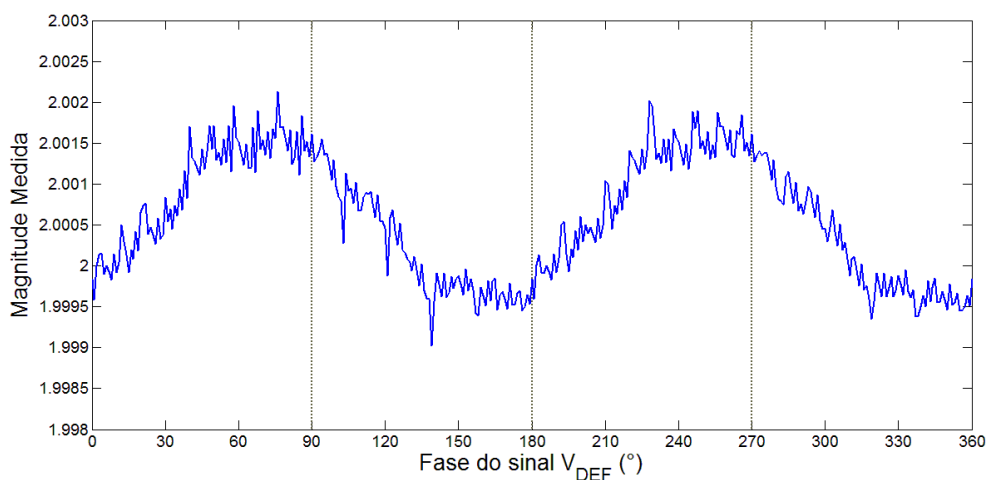


Figura 86 - Resposta de magnitude do modelo do demodulador de quadratura com detector de componente CC para o intervalo $[0^\circ, 360^\circ]$ com passo de 1°

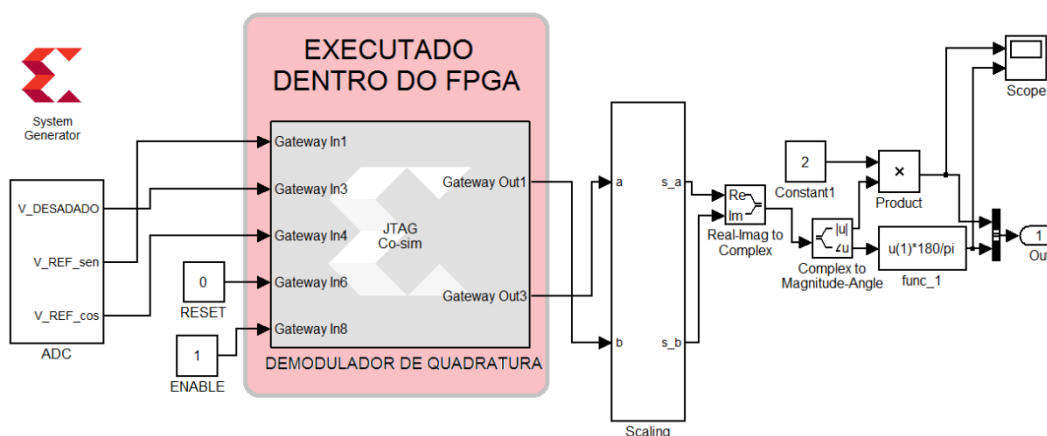


Observa-se nas curvas de resposta obtidas pelo modelo do demodulador de quadratura com detectores de componente CC, que a resposta de fase está de acordo com a resposta da função arco-tangente de dois quadrantes e que o valor de resposta de magnitude variou em $\pm 0,5$ mV, indicando um erro de $\pm 0,025\%$.

4.4.3 VALIDAÇÃO EM *HARDWARE*

O modelo final utilizado para a co-simulação em *hardware* do demodulador de quadratura é mostrado na Figura 87. Todo o código VHDL está compilado dentro do bloco “Demodulador de Quadratura”, sendo que o sinal a ser testado é controlado por um script MATLAB para a varredura do intervalo de defasagem do sinal defasado.

Figura 87 - Modelo do demodulador de quadratura para validação via co-simulação em *hardware*



Na co-simulação em *hardware*, estipulou-se um tempo total de simulação equivalente a 20 ciclos do sinal defasado, sendo que a convergência é obtida após 3 ciclos deste sinal. A varredura do intervalo de defasagem foi executada para todo o intervalo de $[0^\circ, 360^\circ]$ com passo de 1° . Todos os valores estimados pelo modelo durante os últimos 17 ciclos do sinal de referência, foram utilizados para o cálculo do valor médio da resposta do modelo. A curva de resposta de fase obtida através da co-simulação de *hardware* é mostrada na Figura 88 e a curva de resposta de amplitude obtida é mostrada na Figura 89.

Figura 88 - Resposta de fase obtida através da co-simulação em *hardware* do demodulador de quadratura

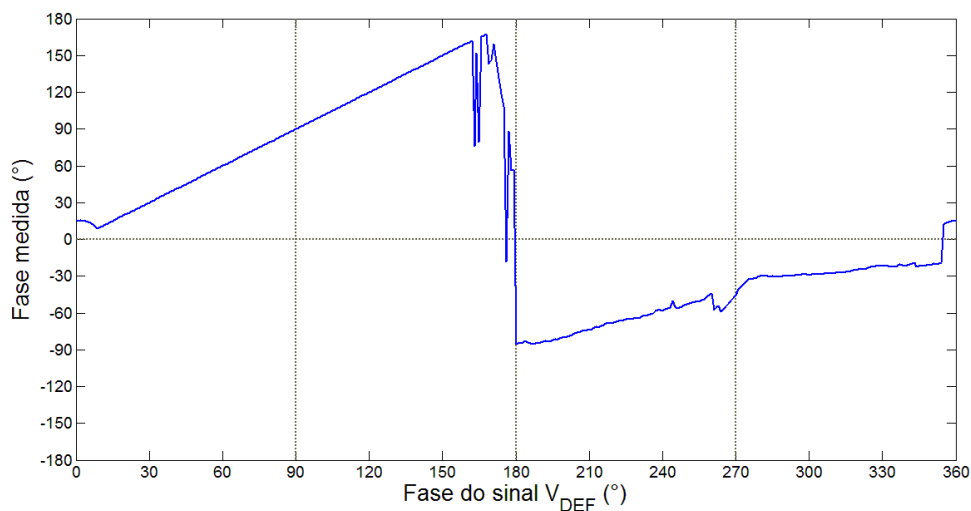
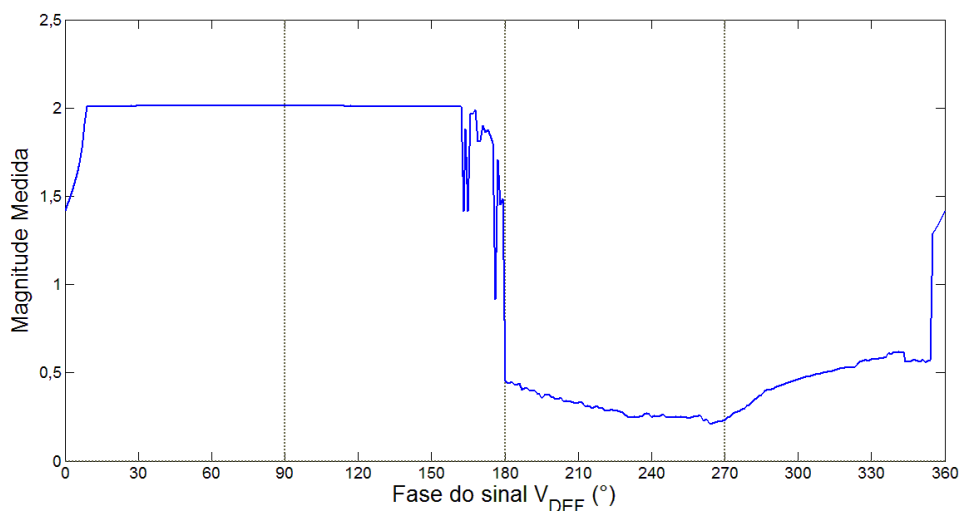


Figura 89 - Resposta de amplitude obtida através da co-simulação em *hardware* do demodulador de quadratura



Os resultados obtidos através da co-simulação em *hardware* indicam que o demodulador de quadratura obteve a resposta correta apenas para o intervalo de defasagem de $[10^\circ, 160^\circ]$. Foram testadas várias configurações do *solver* do Simulink sem melhora na resposta da co-simulação em *hardware*.

4.5 MEDIÇÃO DIRETA DIGITAL

A medição direta digital deve incluir na sua implementação em código VHDL tanto um detector de magnitude quanto um detector de fase. O detector de magnitude implementado em VHDL para a medição direta digital é uma variação da topologia do detector de magnitude apresentado na Figura 31. Na versão implementada, assumiu-se que o sinal medido não tem componente CC, variando sempre com valores de pico positivos e valores de vale negativos

Conceitualmente, a versão implementada do detector de magnitude opera detectando se a variação do sinal de entrada está ocorrendo no semiciclo positivo ou no semiciclo negativo. Quando a variação ocorre no semiciclo positivo, se o sinal de entrada for maior que o valor armazenado na variável “maior valor”, essa variável é atualizada com o valor de entrada. Caso contrário, o valor armazenado é atualizado no *buffer* de saída. De forma semelhante, quando a variação ocorre no semiciclo negativo, se o valor de entrada é menor que o valor armazenado na variável “menor valor”, essa variável é atualizada com o valor de entrada, caso contrário, o valor armazenado é atualizado no *buffer* de saída. Toda vez que ocorre uma passagem por zero, o valor da variável “menor valor” é atualizado para o maior valor positivo possível e o valor da variável “maior valor” é atualizado para o menor valor negativo possível. A detecção da variação do sinal de entrada é executada a cada rampa de subida do sinal de *clock* de amostragem, o qual foi arbitrado como 2 MHz.

Considerando-se a mesma faixa operacional testada para o demodulador de quadratura, a entrada do sinal de defasado no detector de magnitude deve variar na faixa de -1 V a $+1\text{ V}$. Uma vez que não é utilizada aritmética de ponto flutuante na topologia implementada em VHDL, optou-se por utilizar um fator de *scaling* igual a 1000. Dessa forma, a faixa numérica na entrada do detector de magnitude é de -1000 a $+1000$. Para se manter uma margem numérica para testes, optou-se pela utilização da representação numérica de complemento de dois com 16 *bits*. Logo, o maior valor positivo possível de ser representado é $+32.767$ e o menor valor negativo possível de ser representado é -32.768 .

O sinal de referência necessário para o detector de fase é uma onda quadrada com *duty-cycle* próximo de 50% e de frequência 50 kHz. É essencial que o sinal de referência seja gerado sempre com o semiciclo positivo antes do semiciclo negativo, caso isso não seja obedecido, o sinal de referência irá ser visto pelo detector como um sinal defasado, resultando em uma contagem de pulsos errada.

Outro fator crítico para a implementação do detector de fase é a frequência de *clock* da geração do trem de pulsos de contagem. Quanto mais precisa a geração, melhor será a precisão do detector de fase. Para a resolução angular de $0,1^\circ$, as equações (75) e (76) resultaram em uma frequência de geração de trem de pulsos de 180 MHz. Logo, a frequência de *clock* do FPGA deve ser maior que 180 MHz. Para a simulação no System Generator, identificou-se que a frequência de *clock* do FPGA deve obedecer o critério de Nyquist para a maior frequência a ser processada pelo *hardware* do FPGA. Dessa forma, uma vez que a frequência de *clock* para a geração do trem de pulsos de contagem é a maior frequência envolvida, o *clock* do FPGA foi definido como 360 MHz.

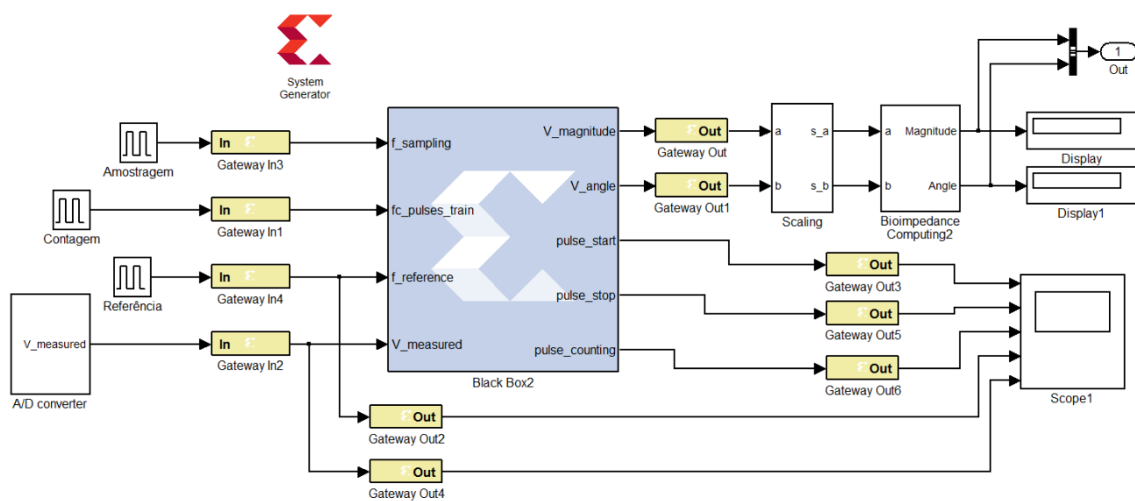
4.5.1 VERIFICAÇÃO DO CÓDIGO VHDL

O código VHDL completo da topologia conjugada de detecção de fase e de magnitude é composta por vários subcomponentes que foram escritos em arquivos separados. Um arquivo *top level* é utilizado para estanciar todos esses componentes dentro de uma topologia completa. Esse arquivo *top level* foi utilizado para se introduzir apenas um único bloco “Black Box” no modelo Simulink / System Generator a ser implementado. O modelo completo é mostrado na Figura 90.

Internamente, o detector de fase faz uso de duas instâncias de subcomponentes do tipo “short pulse” os quais desempenham o papel de monoastáveis e de uma instância de um *flip-flop* RS, modificado para evitar erro no estado proibido. O detector de magnitude, internamente, faz uso de duas instâncias de um subcomponente denominado “ampop_comparator”. Essas

instâncias são utilizadas para determinar se o novo valor amostrado e o último valor amostrado, são positivos, negativos ou iguais a zero. Um sinal lógico distinto é gerado para cada uma das situações. Esses sinais são então utilizados internamente pelo detector de magnitude para detectar se ocorreu alguma passagem por zero, e se o sinal de entrada está variando em um semiciclo positivo ou em um semiciclo negativo.

Figura 90 - Modelo System Generator da medição de fase pela contagem de pulsos



Para a simulação, utilizou-se um tempo total de simulação equivalente a 20 ciclos do sinal defasado e passo de 1° . Considerando que a convergência do modelo ocorre após 3 ciclos completos do sinal defasado, estipulou-se para a simulação, que a resposta do modelo é o valor médio obtido através da média de todos os valores estimados pelo modelo durante os últimos 17 ciclos do sinal defasado. O *solver* utilizado é o solver de passo variável “Discrete (No Continuous States)” com passo máximo de simulação de um quarto do período de uma frequência de 4 MHz. A curva de resposta de fase obtida é mostrada na Figura 91 e a curva de resposta de magnitude obtida é mostrada na Figura 92.

Figura 91 - Resposta de fase do modelo System Generator da técnica de contagem de pulsos

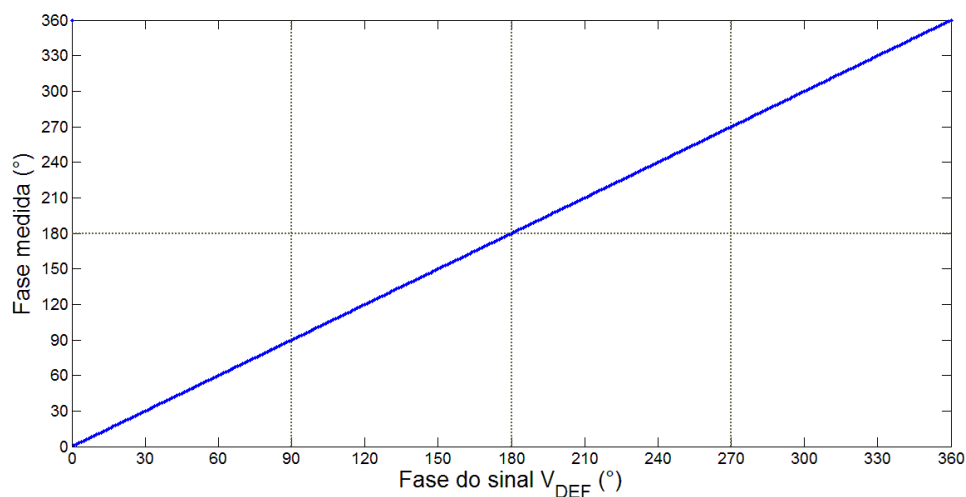
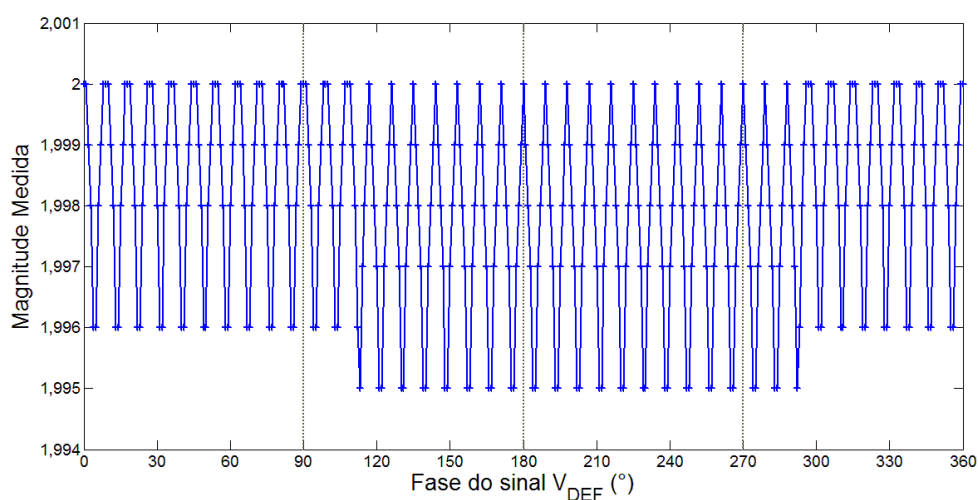


Figura 92 - Resposta de amplitude do modelo System Generator da técnica de contagem de pulsos

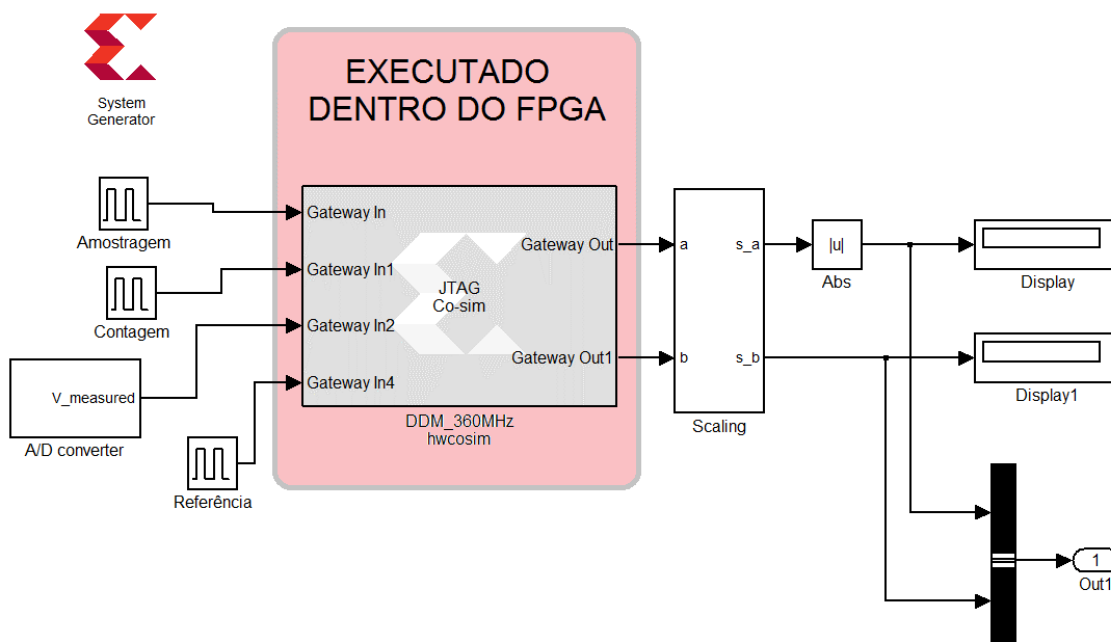


A variação da resposta de fase estimada pelo MATLAB foi de $-0,1^\circ$, sendo que a resposta de fase obtida pelo modelo está de acordo com o resultado obtida pelo modelo Simulink simulado no capítulo 4. A resposta de magnitude apresentou uma variação de -5 mV o que corresponde a uma variação de $-0,25\%$.

4.5.2 VALIDAÇÃO EM *HARDWARE*

O modelo final utilizado para a co-simulação em *hardware* da técnica de contagem de pulsos é mostrado na Figura 87. Todo o código VHDL está compilado dentro do bloco “DDM_360MHz”, sendo que o sinal gerado a ser testado é controlado por um script MATLAB para a varredura do intervalo de defasagem do sinal defasado.

Figura 93 - Modelo final para a co-simulação em *hardware* da técnica de medição de fase por contagem de pulsos



Na co-simulação em *hardware*, estipulou-se um tempo total de simulação equivalente a 20 ciclos do sinal defasado, sendo que a convergência foi obtida após 3 ciclos deste sinal. A varredura do intervalo de defasagem foi executada para todo o intervalo de $[0^\circ, 360^\circ]$ com passo de 10° . Todos os valores estimados pelo modelo durante os últimos 17 ciclos do sinal de referência, foram utilizados para o cálculo do valor médio da resposta do modelo. O *solver* utilizado é um *solver* de passo variável discreto. A curva de resposta de fase obtida através da co-simulação de *hardware* é mostrada na Figura 94 e a curva de resposta de amplitude obtida é mostrada na Figura 95.

Figura 94 - Resposta de fase da co-simulação em *hardware* da técnica de contagem de pulsos

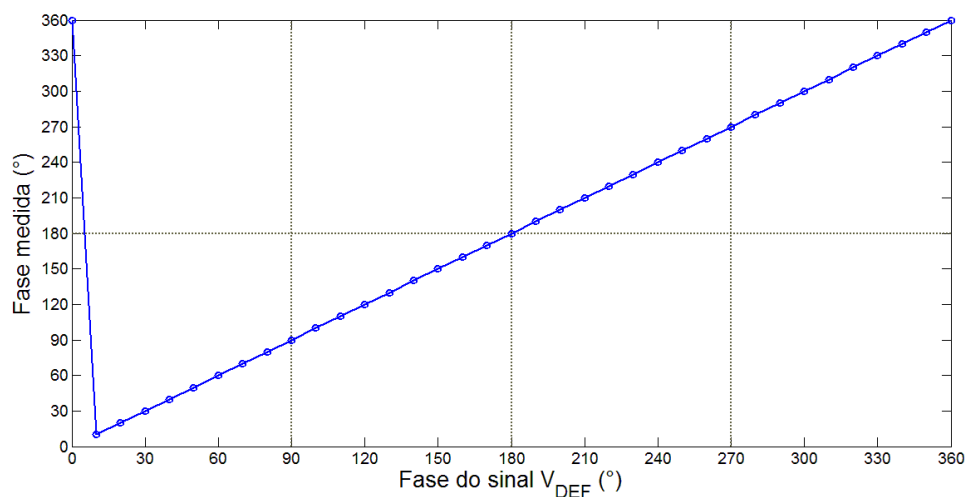
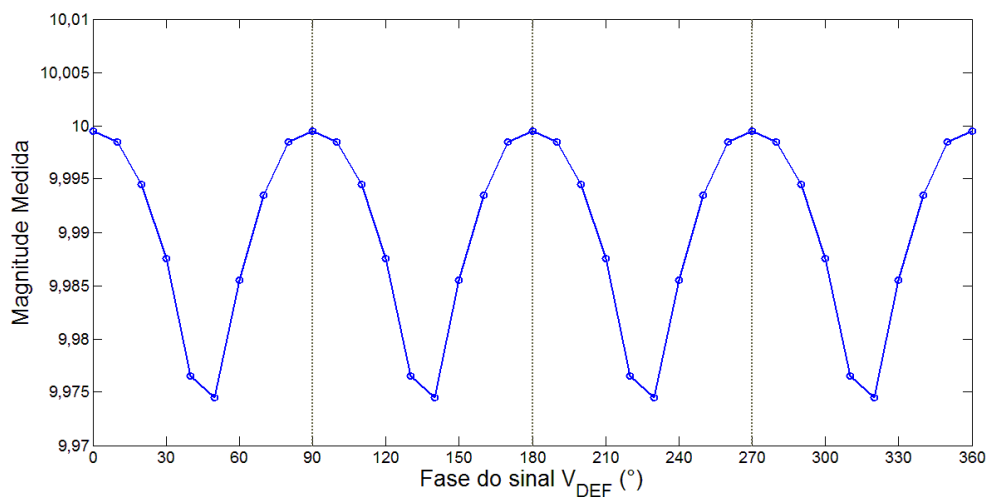


Figura 95 - Resposta de amplitude da co-simulação de *hardware* da técnica de contagem de pulsos



A variação da resposta de fase estimada pelo MATLAB foi de $-0,1^\circ$, sendo que a resposta de fase obtida pelo modelo está de acordo com a simulação do modelo System Generator. A amplitude do sinal testado foi de 10 volts resultando em uma variação de até -25 mV o que corresponde a uma variação de $-0,25\%$.

4.6 COMPARAÇÃO DAS IMPLEMENTAÇÕES EM VHDL E A LITERATURA

Conforme os resultados obtidos através da verificação do código implementado em VHDL, o demodulador de quadratura simulado nesse trabalho pode ser comparado com o demodulador de quadratura apresentado em Martinez-Nolasco et al. (2012) e o medidor de fase por contagem de pulso pode ser comparado com o medidor apresentado por Rathore (2003). A comparação é mostrada na Tabela 11 e foi feita em termos de erro de medição de fase e frequência de *clock* do FPGA.

Tabela 11 - Comparação entre as topologias implementadas em VHDL e as topologias reportadas na literatura

Parâmetro	Demodulador de Quadratura		DDM – Contagem de Pulsos	
	Martinez-Nolasco et al. (2012)	Topologia Proposta	Rathore (2003)	Topologia Proposta
Erro	Re (1% - 3%) Im (1% - 7%)		± 1 pulso	± 1 pulso
Erro de fase	1° a 6°	$\pm 0,65^\circ$	$\pm 0,1^\circ$	$\pm 0,1^\circ$
Frequência de <i>Clock</i>	50 MHz	100 MHz		360 MHz
Frequência de Pulsos de Contagem	-	-	-	180 MHz

Observa-se através da comparação, que o demodulador de quadratura com a inserção do detector de componente CC apresentou um erro de medição de fase inferior e estável ($\pm 0,65^\circ$), quando comparado ao demodulador de quadratura apresentado por Martinez-Nolasco et al. (2012).

O medidor de fase por contagem de pulso implementado neste trabalho apresentou o mesmo tipo de erro reportado por Rathore (2003), ficando este erro na faixa de mais um ou menos um pulso de contagem. Como o medidor de fase

por contagem de pulso foi implementado para a resolução de $0,1^\circ$ para cada pulso de contagem, o erro obtido foi de $\pm 0,1^\circ$.

4.7 COMPARAÇÃO DO CONSUMO DE RECURSOS DO FPGA ENTRE O DQ E O DDM

A comparação dos recursos do FPGA utilizados por cada uma das topologias é apresentada na Tabela 12. O método DDM apresentado na tabela é composto tanto pelo medidor de fase por contagem por pulsos quanto o detector de magnitude. O demodulador de quadratura comparado é a topologia com a inserção dos detectores de componente CC na saída dos filtros.

Tabela 12 - Comparação do consumo de recursos no FPGA

Utilização Lógica	Disponível	Demodulador de Quadratura	DDM
Slice Registers	28.800	3674	105
Slice LUTs	28.800	2982	174
Fully LUT-FF Pairs	204	606	75
BUFG/BUFGTCTRL	32	3	4
DSP48Es	48	10	-

Observa-se que a técnica de medição direta digital (DDM) consome consideravelmente menos recursos que o demodulador de quadratura. Com base nesse critério, a técnica DDM acaba sendo apontada como o melhor método em termos de menor consumo de pastilha de silício. Porém, uma vez que esse método pode ser bastante suscetível ao ruído, deve ser executada uma análise da área de pastilha de silício consumida por um bloco analógico de condicionamento de sinal, assim como a de um filtro digital na entrada do DDM.

5. CONCLUSÕES E TRABALHOS FUTUROS

Durante o desenvolvimento deste trabalho foram testados sete métodos de medição de fase através do Simulink. Quatro métodos trabalham com nuvem de pontos, sendo da categoria de técnicas que possuem uma imunidade a ruído intrínseca e três métodos são métodos de medição direta, os quais não apresentam imunidade a ruído. Os métodos do demodulador de quadratura e o método da medição digital direta por contagem de pulsos foram selecionados para implementação em código VHDL. Ambas as topologias seguiram o processo de verificação do código VHDL através de emulação via *software* e de validação do código via co-simulação em *hardware*. Com base na análise dos resultados obtidos na simulação teórica, na verificação de código e na validação via co-simulação em *hardware*, foram feitas as conclusões e observações apresentadas nesta seção.

5.1 CONCLUSÕES

5.1.1 A RESPEITO DOS MÉTODOS DE NUVENS DE PONTOS

Assumiu-se que esses métodos conseguem obter imunidade intrínseca a ruído conforme especificado pelos autores consultados durante a revisão de referências. Nenhum teste detalhado de resposta ao ruído foi executado para comparar ou classificar o grau mínimo de imunidade a ruído dos métodos de nuvem de pontos. O único teste de resposta ao ruído executado foi para um único sinal defasado em 30° conforme mostrado no apêndice A. Para a verificação das técnicas os resultados teóricos foram comparados quanto à resposta em condições ideais.

Os resultados obtidos para o método do ajuste de elipse mostraram que o mesmo é falho para medir valores de fase próximos de 0° e de 180° . Ainda, verificou-se que esse método possui a maior complexidade matemática e a maior

demanda de quantidade de recursos computacionais, dentro todos os métodos testados.

Segundo o critério de demanda de recursos computacionais, o método do ajuste da senóide se apresentou como o segundo método com maior demanda de recursos. Por outro lado, apesar da demanda de recursos computacionais, esse método não apresenta complexidade matemática grande, sendo que a matriz de coeficientes calculados a partir do seno e cosseno podem ser armazenados diretamente dentro de uma ROM. Isso pode permitir uma implementação vantajosa, principalmente se for possível a utilização de um menor número de amostras do que as 256 que foram testadas.

O método da Transformada Direta de Fourier com algoritmo Radix-4 apresentou falhas durante a simulação, não conseguindo identificar de forma constante, o valor da fase do sinal medido no quadrante correto. O valor retornado foi o valor equivalente da fase nos quatro quadrantes cartesianos possíveis. Considerando-se o valor que mais se repete como o resultado correto, os resultados obtidos para 1024 amostras e 256 apresentaram praticamente nenhuma oscilação de quadrante no valor final, porém, o valor se apresenta 180° defasado. Esse método é indicado apenas para aplicações que necessitam de uma análise espectral para se obter a informação desejada.

Os resultados obtidos para o método do demodulador de quadratura, mesmo para uma topologia com filtros FIR de ordem 40, mostraram que o demodulador de quadratura apresentou a menor demanda de recursos computacionais dentre todos os métodos de nuvem de pontos testados. A utilização de uma frequência de amostragem de 2 MHz para um sinal de entrada de 50 kHz, resultou em 40 amostras e o teste da ordem do filtro mostrou que um filtro passa-baixa FIR de ordem 40 apresenta a melhor resposta para as especificações testada.

5.1.2 A RESPEITO DOS MÉTODOS DE MEDIÇÃO DIRETA

Dentro da categoria dos métodos de medição direta, a análise da quantidade de operações matemáticas e pulsos de *clock* necessários mostrou

que o método de medição direta pela técnica de Lissajous é o método de medição direta com maior consumo de recursos computacionais. Pela natureza como é feita a medição nesse método, verificou-se que esse método necessita de grande velocidade de amostragem para permitir que amostra da amplitude do sinal medido tenha o valor correto no momento exato que ocorre a passagem por zero do sinal de referência. Se o período de amostragem for maior que o período mínimo da discretização angular desejada, os erros de estimativa de fase serão bastante grosseiros. A presença de ruído pode agravar ainda mais esse fato.

O método da contagem de pulsos se mostrou como o método com a segunda maior demanda de recursos computacionais. Esse valor foi estimado para um valor defasado em 360° , pois a demanda de recursos é diretamente proporcional ao número de pulsos que devem ser contados para a defasagem do sinal medido. Para um ângulo de defasagem próximo de 0° , a demanda de recursos computacionais para o cálculo da fase é praticamente nula. Apesar do erro ser baixo, da ordem de mais ou menos um pulso de contagem, o que implica em um erro na ordem da resolução angular implementada para a contagem dos pulsos, a grande desvantagem desse método é que a frequência de *clock* necessária para gerar o trem de pulsos de contagem tem que ser elevada.

O método da medição através da modulação de largura de pulso (PWM) não permite comparação direta da demanda de recursos computacionais, uma vez que o mesmo não efetua operações aritméticas, sendo estas, o critério para a definição da eficiência computacional. Se o filtro implementado for um filtro digital, pode-se obter um valor comparativo de demanda de recursos computacionais.

Se o filtro for implementado na forma digital, deve-se acrescentar na saída do filtro um bloco lógico para calcular o valor médio do sinal obtido, compensando-se dessa forma o *ripple* característico do filtro, o qual é inevitável. A partir do valor médio obtido na saída do filtro, pode-se interpolar de forma digital uma curva de fase em função do valor na saída do filtro, permitindo a medição direta de um sinal representativo de fase. Essa solução totalmente digital implica necessariamente em uma matemática mais complexa através do uso de *scaling* ou de matemática de ponto flutuante. Consequentemente, a

solução totalmente digital do método PWM será mais complexa matematicamente que o método de medição direta por contagem de pulsos.

5.1.3 A RESPEITO DA VERIFICAÇÃO DO CÓDIGO VHDL

Os testes da implementação VHDL do demodulador em quadratura com emulação via *software* pelo System Generator mostraram um resultado similar ao resultado obtido com o modelo Simulink para simulação teórica do método. Ao contrário do resultado obtido pelo modelo teórico, a implementação em VHDL mostrou uma inserção significativa de *ripple* na harmônica que é metade da frequência fundamental do sinal medido. Um dos motivos para essa ocorrência é a perda numérica através do uso da técnica de *scaling*. Enquanto o Simulink utiliza aritmética de ponto flutuante com números de 64 *bits*, a solução VHDL com *scaling* resultou em números inteiros em complemento de dois com 64 *bits*, porém a aritmética que resultou nesse número de bits foi a multiplicação de dois números inteiros de 32 *bits*.

Devido ao *ripple* encontrado na saída do filtro passa-baixa do demodulador de quadratura, efetuou-se a introdução de um detector de componente CC na saída de cada filtro passa-baixa. O detector de componente CC calcula a média entre o valor de pico positivo e o valor de pico negativo a cada passagem por zero. De certa forma, o comportamento do detector de componente CC é similar a um filtro de média móvel. O resultado obtido com a inserção do detector de componente CC foi a redução da amplitude do *ripple* para 6% do seu valor da amplitude original.

A verificação do código VHDL da técnica de contagem de pulsos apresentou o resultado e o desvio de medida condizente com o que foi observado na simulação do modelo Simulink do método teórico. O erro de medição de fase na emulação via *software* do código VHDL foi de menos um pulso de contagem.

O detector de magnitude de pico utilizado para a medição da amplitude do sinal defasado apresenta um erro que varia de acordo com a frequência de amostragem utilizada para o mesmo. Verificou-se que para a frequência de

amostragem de 2 MHz, a precisão angular para a fase do sinal de 50 kHz é de 9° , resultando em uma variação de até 5 mV na medição de amplitude. Esse erro é menor que 1%, e pode ser reduzido aumentando-se a frequência de amostragem. Para uma frequência de amostragem resultante numa precisão de $0,1^\circ$, o valor da magnitude pode ser considerado numericamente constante, apresentando a magnitude no valor correto.

5.1.4 A RESPEITO DA VALIDAÇÃO EM *HARDWARE*

A co-simulação em *hardware* do demodulador de quadratura apresentou resposta correta apenas para o intervalo de defasagem do sinal medido, correspondente a $[10^\circ, 160^\circ]$. Uma resposta semelhante a essa foi encontrada durante alguns testes de verificação do código VHDL do demodulador de quadratura via emulação em *software*. Identificou-se na emulação via *software*, que esse problema era gerado devido às configurações de passo de tempo do *solver* utilizado. Na emulação via *software*, o problema foi resolvido através da configuração de passo máximo de tempo do *solver*, como sendo um quarto do valor do período de amostragem do filtro. A mesma solução não funcionou para o teste de co-simulação de *hardware*. Foram testados passos de tempo ainda menores, porém, verificou-se que o problema não é relacionado ao código VHDL ou aos coeficientes do filtro, mas sim, a algum problema durante a configuração automática do gerenciamento de troca de dados entre o Simulink e o FPGA.

A co-simulação em *hardware* da técnica de medição direta por contagem de pulsos apresentou uma resposta idêntica à resposta obtida na verificação do seu código VHDL na emulação via *software*. Assim, considera-se que a técnica de medição de fase por contagem de pulso foi validada para todo o intervalo operacional de $[0^\circ, 360^\circ]$.

5.1.5 A RESPEITO DO CONSUMO DE RECURSOS DO FPGA

De acordo com os resultados de consumo de recursos do FPGA obtidos com a síntese do código VHDL e mostrados na tabela Tabela 12, o método de

medição direta digital por contagem de pulsos apresenta um consumo de recursos consideravelmente menor que o método do demodulador de quadratura. Por outro lado, o método da contagem de pulsos necessita de uma frequência de *clock* consideravelmente maior que o método do demodulador de quadratura. Na prática não se pode simplesmente afirmar que o método da medição de contagem de pulsos resultaria em uma topologia com menor utilização de pastilha de silício. Quando os métodos são considerados isoladamente, essa premissa pode ser verdadeira, porém, uma vez que o método de medição direta digital não possui nenhuma imunidade a ruído, uma análise de *trade off* mais cuidadosa deve ser feita levando em consideração a área de silício que seria ocupada por um circuito de condicionamento de sinal na entrada do bloco de medição direta digital.

5.2 TRABALHOS FUTUROS

Os métodos de medição de magnitude e fase de sinais estudados no desenvolvimento dessa pesquisa, são métodos que podem ser utilizados em qualquer aplicação que necessite desse tipo de medição. Ou seja, mesmo que o caso de estudo desta pesquisa tenha sido uma aplicação biomédica baseada na medição de impedância para um sinal único de 50 kHz, os métodos continuam válidos para qualquer aplicação, bastando apenas se especificar um novo envelope operacional e valores desejados para resolução angular e resolução de magnitude. Tendo em vista a aplicabilidade dessas técnicas, esta seção apresenta possibilidades de trabalhos futuros tanto para o caso de aplicação estudada como para a continuidade da pesquisa acerca dos métodos apresentados, buscando-se novas aplicações.

5.2.1 RELACIONADOS AO PROCEDIMENTO DE VERIFICAÇÃO E VALIDAÇÃO DO CÓDIGO VHDL

A utilização do System Generator trouxe uma grande flexibilidade para o desenvolvimento de aplicações baseadas em FPGA. O fato de não se ter

conseguido validar em *hardware* todo o intervalo de resposta de $[0^\circ, 360^\circ]$ para o demodulador de quadratura, indica a necessidade de se aprofundar uma investigação da razão do comportamento inválido da co-simulação de *hardware*, pois desenvolvimentos futuros podem ser impactados pelo mesmo tipo de problema.

Nesse sentido, pode-se também explorar a utilização de outros recursos da placa do próprio FGPA para se gerar um sinal de teste e mostrar a resposta que é enviada para a saída do FPGA. Apesar de ser desvantajoso do ponto de vista de armazenamento de resultados para comparação teórica, essa aproximação permitiria uma validação manual em *hardware* tornando o ciclo de desenvolvimento de uma aplicação em FPGA (projeto conceitual \rightarrow projeto detalhado \rightarrow verificação do código VHDL \rightarrow validação da síntese em *hardware*) menos dependente de uma única ferramenta.

Como consequência de se implementar um método de visualização de resultados na própria placa do FPGA para validação manual em *hardware*, acaba-se tendo a oportunidade de se implementar um módulo em VHDL para o cálculo do valor de fase e do valor de magnitude, de forma que esses valores possam ser apresentados diretamente na forma dos valores de fase e magnitude medidas.

5.2.2 RELACIONADOS AO DEMODULADOR DE QUADRATURA

O demodulador de quadratura utiliza necessariamente um sinal de referência seno e um sinal de referência cosseno. Nos testes apresentados neste documento foi mostrada apenas a utilização do demodulador de quadratura com a geração externa desses sinais. Um dos resultados obtidos com a pesquisa desse trabalho foi o artigo intitulado ***“A Comparison between Direct Digital Measurement Technique and Digital Quadrature Demodulation for Complex Bioimpedance Measurement Implementation in FPGA”***, o qual foi apresentado no congresso “Iberchip XXI Workshop (IWS’2015)”. Este artigo apresenta a geração dos sinais de referência através de *look up tables* de seno e cosseno, sendo que os sinais de referência tem amplitude máxima unitária e

fase de zero graus. A discretização utilizada nessas tabelas foi de $3,6^\circ$, o que não permite uma resolução angular adequada para as especificações expressas neste documento. Dessa forma, faz-se necessária a exploração de tabelas de seno e cosseno com maior resolução angular.

A inserção da geração dos sinais de referência como parte da topologia do demodulador de quadratura traz vantagens por permitir uma melhor garantia de que os sinais de referência realmente sincronizados. Isso ainda é bastante interessante quando se lida com aplicações de medição de impedância, pois o mesmo componente que faz a medição de magnitude e fase pode gerar o sinal que controla a fonte de injeção do sinal na amostra que está sendo testada. Assim, a respeito da geração dos sinais de referência, identifica-se como pontos a serem explorados os seguintes aspectos: controle da sincronização de sinais de referência gerados externamente de modo a evitar erros de medida, comparar o uso de tabelas de seno e cosseno com alta resolução angular frente ao uso do DSP do FPGA para geração dos sinais de referência, e por último, verificar se haveria ganho na utilização e/ou desenvolvimento de um módulo DDS (*Direct Digital Synthesis*) interno ao componente de medição de magnitude e fase.

O demodulador de quadratura testado nesta pesquisa utiliza filtros passa baixa digitais do tipo FIR. Pode-se explorar futuramente, o uso de outros filtros como um filtro do tipo IIR, e até mesmo, se efetuar uma comparação para verificar se para uma dada resolução angular e de magnitude desejada, se a topologia digital é melhor que a topologia analógica, tanto em termos de desempenho como em termos de consumo de área de pastilha de silício.

5.2.3 RELACIONADOS AO MÉTODO DDM

Uma vez que o método de medição de fase por contagem de pulsos aliado ao detector de magnitude para medição dos picos positivos e picos negativos foi validada para todo o intervalo de interesse de $[0^\circ, 360^\circ]$, a topologia desenvolvida para esse método é uma boa candidata para se completar o ciclo de desenvolvimento de um ASIC. Dessa forma, pode-se explorar a geração do *layout* do circuito integrado a partir do código VHDL por uma ferramenta como o

CADENCE, e depois de verificar o comportamento elétrico, proceder com a fabricação do protótipo desse ASIC. Caso se opte por seguir com o desenvolvimento de um ASIC, a topologia do DDM é simples, principalmente se o detector de magnitude receber um sinal digitalizado cuja componente CC é 0 volts. Porém, apesar da simplicidade, o projeto de um ASIC envolve a exploração da integração de um ADC para indicar qual é a amplitude do sinal de entrada no componente de DDM.

A topologia mais simples do DDM permite que se explore outras formas de projeto de circuitos que não sejam o uso de linguagens de especificação de *hardware* como o VHDL e o Verilog. Essa topologia e a sua integração com outros blocos funcionais de microeletrônica.

5.2.4 RELACIONADOS AOS MÉTODOS TEÓRICOS PESQUISADOS

Apesar da implementação em código VHDL de alguns métodos ter sido descartada nessa pesquisa, devido à complexidade matemática dos mesmos, seria bastante interessante que se prosseguisse no futuro com uma solução de implementação em VHDL para cada método identificado. O desenvolvimento de um trabalho como esse permitiria investigar melhor alguns parâmetros como o efeito do número de amostras realmente necessárias para o uso do método, assim como a busca por soluções que permitam o uso de *undersampling*.

O método de medição de fase e magnitude através de PWM é um método que deveria ser explorado no futuro. O uso dessa técnica é essencial no desenvolvimento de PLLs e a topologia do extrator de fase proposta por Lawson e Brown (1981) foi implementada recentemente por Ausín et al. (2012) em tecnologia CMOS de 350 nm para a medição de impedância bioelétrica, com bons resultados, tendo erro de 0,4°. Uma comparação entre a solução analógica empregada por Ausín et al. (2012) e uma solução totalmente digital poderia trazer ganhos para melhoria de topologia PWM visando a obtenção de uma maior resolução angular.

Tendo em vista que todos os métodos pesquisados são métodos gerais aplicáveis a qualquer problema de medição de magnitude e de fase de sinais, um passo futuro importante seria realizar um conjunto de simulações bem estruturadas para se determinar quais são os menores valores de razão sinal ruído possíveis de se obter em cada método. Adicionalmente uma análise de sensibilidade poderia ser feita, permitindo a parametrização desses modelos de acordo com características de projeto, permitindo um desenvolvimento de uma topologia mais robusta do ponto de vista elétrico.

Ainda, uma vez que se saiba como os parâmetros se relacionam com o ruído, pode-se desenvolver um modelo que permite levar em conta fatores intrínsecos de *hardware* como ruído térmico e interferência eletromagnética. Em aplicações críticas como aplicações aeroespaciais, militares e implantes biomédicos, o conhecimento do impacto do ruído de hardware é essencial.

Caso se prossiga com o desenvolvimento desses modelos mais robustos, abre-se uma possibilidade bastante interessante de reuso do conhecimento. Através do uso da linguagem VHDL e de uma linguagem de modelagem elétrica como a VHDL-AMS (*hardware description language and mixed-signal extensions*) compatível com as ferramentas de projeto de circuito integrado como o CADENCE, pode-se criar uma biblioteca de componentes com topologias geradas de forma automática ou semi-automática para serem integradas dentro de soluções ASIC para medição de fase e magnitude.

A geração dessa automatização envolve necessariamente o processo de verificação de modelos elétricos e validação dos mesmos frente a tempo de vida, erros de operação e envelopes operacionais extremos. O trabalho realizado a longo prazo, permitiria não só se desenvolver processos de projeto de circuitos integrados visando a robustez e confiabilidade, mas também, a criação de uma biblioteca de conhecimento e modelos nos moldes da MBSE (*Model-based System Engineering*).

5.2.5 RELACIONADOS À BIOIMPEDÂNCIA

Uma vez que o caso de estudo da pesquisa realizada é o emprego dos métodos de medição de fase e magnitude de sinais para a medição de bioimpedância SFBIA de 50 kHz, seria bastante valioso no futuro, integrar o FPGA com o código VHDL sintetizado já embarcado, dentro de uma solução de *hardware* de medição de valores bioimpedância. Durante o desenvolvimento dessa pesquisa, não haviam módulos de ADC disponíveis para realizar a integração da placa do FPGA junto a circuitaria analógica. Mas uma vez sendo possível realizar essa integração, os métodos de medição de fase e magnitude validados poderiam ser testados em uma situação real ao invés de valores emulados matematicamente pelo Simulink.

Conforme apontado no capítulo 1, a técnica de bioimpedância SFBIA não é a única técnica de bioimpedância existente, pelo contrário, existem mais três técnicas e todas elas trabalham com múltiplas frequências. Sendo assim, uma evolução natural do desenvolvimento de solução VHDL para medição de fase e magnitude seria a expansão do código VHDL desenvolvido nesta pesquisa para abranger a medição de impedância em diversas frequências.

5.2.6 RELACIONADOS À ARITMÉTICA DE PONTO FLUTUANTE

Uma última oportunidade de desenvolvimento futuro que vale a pena ser citada é o desenvolvimento de uma unidade lógica aritmética para aritmética de ponto flutuante. Nesta pesquisa foi estudada a FPU100 versão 19 por possuir código aberto. Algumas tentativas de otimização do código da FPU foram feitas, mas devido ao tempo e ao volume de trabalho, a otimização de código foi abandonada. Dessa forma, fica a perspectiva futura, do desenvolvimento de uma versão customizada de uma FPU que possa realizar as operações aritméticas de maneira mais rápida, o que é muito importante em aplicações com alta taxa de amostragem.

REFERÊNCIAS

AL-ERYANI, J. FPU100. Version 19. Disponível em: <<http://opencores.org/project,fpu100>>. Acesso em: 16/7/2014.

AL-RAWI, A. H. M.; IBRAHIM, W. M. A. B. W.; MIRZA, E. H. DC feedback for wide band frequency fixed current source. **Journal of Electrical Bioimpedance**, v. 4, n. 1, p. 33–37, 2013.

AMARAL, C. E. F.; LOPES, H. S.; ARRUDA, L. V; et al. Design of a complex bioimpedance spectrometer using DFT and undersampling for neural networks diagnostics. **Medical engineering & physics**, v. 33, n. 3, p. 356–61, 2011.

AMARAL, C. E. F.; BRISCHWEIN, M.; WOLF, B. Multiparameter techniques for non-invasive measurement of blood glucose. **Sensors and Actuators B: Chemical**, v. 140, n. 1, p. 12–16, 2009.

AROOM, K. R.; HARTING, M. T.; COX, C. S.; et al. Bioimpedance analysis: a guide to simple design and implementation. **The Journal of surgical research**, v. 153, n. 1, p. 23–30, 2009. Elsevier Inc.

AUSÍN, J. L.; RAMOS, J.; DUQUE-CARRILLO, J. F.; TORELLI, G. A High Dynamic Range Wideband CMOS Phase Angle Detector for Bioimpedance Spectroscopy. Electronics, Circuits and Systems (ICECS), 2012 19th IEEE International Conference on. **Anais...** . p.376–379, 2012. Seville: IEEE.

BELMONT, B.; DODDE, R. E.; SHIH, A. J. Impedance of tissue-mimicking phantom material under compression. **Journal of Electrical Bioimpedance**, v. 4, n. 1, p. 2–12, 2013.

BERTEMES-FILHO, P.; VALICHESKI, R.; PEREIRA, R. M.; PATERNO, A S. Bioelectrical impedance analysis for bovine milk: Preliminary results. **Journal of Physics: Conference Series**, v. 224, p. 012133, 2010.

BIELA, A.; WATKINSON, M.; MEIER, U. C.; et al. Disposable MMP-9 sensor based on the degradation of peptide cross-linked hydrogel films using electrochemical impedance spectroscopy. **Biosensors & bioelectronics**, v. 68C, p. 660–667, 2015.

BIODYNAMICS-CORPORATION. **Clinician Desk Reference for BIA Testing Table of Contents**. Biodynamics Corporation, 2014.

BLACK&DECKER. Manual de Instruções BK50P - Balança Digital Body Control. , 2012. Uberaba: Black & Decker do Brasil Ltda.

BONILLA, A.; L., R. J. V.; LENZI, K. G.; MELONI, L. G. P. Design and Implementation of Fast Fourier Transform Algorithm in FPGA. XXX SIMPÓSIO

BRASILEIRO DE TELECOMINCAÇÕES - SBrt'12. **Anais...** . p.13–16, 2012. Brasília, DF: Sociedade Brasileira de Telecomunicações.

CARDOSO, A. DOS S. **Desenvolvimento e Implementação de um Sintetizador de Frequência CMOS utilizando Sistema Digital**, 2009. UNESP Universidade Estadual Paulista.

CERÁVOLO, F. A. **PLATAFORMA DE MEDIÇÃO DE GORDURA CORPORAL POR BIOIMPEDÂNCIA**, 2012. Universidade Federal do Paraná.

DONOVAN, M.; ROMERO, A. Fixed Point Radix-4 FFT Fixed Point Radix-4 FFT - File Exchange - MATLAB Central. Disponível em: <<http://www.mathworks.com/MATLABcentral/fileexchange/22326-fixed-point-radix-4-fft>>. Acesso em: 23/5/2014.

FAT. FIGURAS DE LISSAJOUS E MEDIDAS DE DEFASAGEM. Disponível em: <<http://www.fat.uerj.br/intranet/disciplinas/Eletricidade V/6. Figuras de Lissajous e medidas de defasagem.doc>>. Acesso em: 10/5/2014.

FRITSCH, A. R.; GOERGEN, D. L.; SILVA, D. T. DA; SOUZA, K. E. M. DE; ALBRECHT, R. Efeitos Fisiológicos do Choque Elétrico no Corpo Humano. **Jornal Mundo Físico - UDESC**, v. V, n. 25, 2008. Joinville.

GRIMNES, S.; MARTINSEN, Ø. G. **BIOIMPEDANCE AND BIOELECTRICITY BASICS**. Second Edi ed. London: Elsevier Ltd., 2008.

HALIR, R.; FLUSSER, J. Numerically stable direct least squares fitting of ellipses. WSCG' 98 6-th International Conference in Central Europe on Computer Graphics, Visualization'98 Proceedings. **Anais...** . p.125–132, 1998. University of West Bohemia, Campus Bory, Czech Republic: University of West Bohemia, Campus Bory, Czech Republic.

HOLDER, D. S. **Electrical Impedance Tomography: Methods, History and Applications**. IOP Publishing Ltd, 2004.

HUSSAIN, M. W. **Low Power Implantable ASIC for Bio-Impedance Measurements**, 2012. Royal Institute of Technology. Stockholm, Sweden.

HYNIX, M. A. D. T. **AN110 - D / A Converter by Using PWM Using PWM to Generate Analog Output (8-bit Microcontroller Application Note)**. 2001.

IMPEDMED. Fluid Status / Body Composition BIS is a safe , rapid , noninvasive technique for assessing multiple fluid compartments and body composition. Disponível em: <<http://www.impedimed.com/medical-applications/fluid-status-body-composition/>>. Acesso em: 8/5/2013.

ISAACSON, D. Distinguishability of conductivities by electric current computed tomography. **IEEE transactions on medical imaging**, v. 5, n. 2, p. 91–5, 1986.

KARKI, J. **SLOA024B - Analysis of the Sallen-Key Architecture Application Report**. Dallas, 2002.

KYLE, U. G.; BOSAEUS, I.; DE LORENZO, A. D.; et al. Bioelectrical impedance analysis--part I: review of principles and methods. **Clinical nutrition (Edinburgh, Scotland)**, v. 23, n. 5, p. 1226–43, 2004.

LAWSON, K. D.; BROWN, N. L. **PHASE MEASURING CIRCUIT**. , 1981. USA.

LIN, C.-M.; CHEN, L.-H.; CHEN, T.-M. The development and application of an electrical impedance spectroscopy measurement system for plant tissues. **Computers and Electronics in Agriculture**, v. 82, p. 96–99, 2012. Elsevier B.V.

LORENZO, A. DE; ANDREOLI, A.; MATTHIE, J.; WITHERS, P. Predicting body cell mass with bioimpedance by using theoretical methods: a technological review. **Journal of Applied Physiology**, v. 82, p. 1542–1558, 1997.

LULU, X.; PING, Z.; SONGZHE, C.; LAIJUN, W. In-situ electrochemical impedance spectroscopy measurement of anodic reaction in SO₂ depolarized electrolysis process. **Chemical Engineering and Processing: Process Intensification**, v. 89, p. 70–74, 2015. Elsevier B.V.

MARTINEZ, F. S. **Electrical Bioimpedance Cerebral Monitoring: Fundamental Steps towards Clinical Application by Electrical Bioimpedance**, 2007. CHALMERS UNIVERSITY OF TECHNOLOGY.

MARTINEZ-NOLASCO, C.; DIAZ-CARMONA, J.; PRADO-OLIVAREZ, J.; RAMIREZ-AGUNDIS, A.; PADILLA-MEDINA, A. FPGA Synchronous Phase Detection System for bioimpedance computing applications. **2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)**, p. 362–365, 2012. IEEE.

OLIVEIRA, J. I. S. DE. **Medição de múltiplas fases de nível de líquidos usando filtro adaptativo: técnicas , métodos e simulações**, 2005. UNIVERSIDADE FEDERAL DO MARANHÃO.

PAPEZOVA, S. Signal processing of bioimpedance equipment. **Sensors and Actuators B: Chemical**, v. 95, n. 1-3, p. 328–335, 2003.

PHATUNGTHANE, T.; RUJIJANAGUL, G.; PENGPAT, K.; et al. Dielectric and impedance measurements on (1-x)Ba(Fe_{1/2}Ta_{1/2})O₃-xBa(Zn_{1/3}Ta_{2/3})O₃ ceramics. **Current Applied Physics**, v. 14, n. 12, p. 1819–1824, 2014. Elsevier B.V.

PIROUZPANAH, S.; ERDOĞAN, B.; MORRISON, G. L. Study of hysteresis effects and emulsion properties in watercut measurement using high speed multi-frequency impedance sensor. **Flow Measurement and Instrumentation**, v. 40, p. 1–8, 2014.

PLONSEY, R.; BARR, R. C. **Bioelectricity A Quantitative Approach**. Third Edit ed. New York: Springer Science+Business Media, LLC, 2007.

PORTO, R. W. **Projeto e avaliação de um canal de medição de bioimpedâncias**, 2009. ESCOLA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL.

RAMOS, P. M.; JANEIRO, F. M.; RADIL, T. Comparative Analysis of Three Algorithms for Two-Channel Common Frequency Sinewave Parameter Estimation: Ellipse Fit, Seven Parameter Sine Fit and Spectral Sin Fit. **Metrology and measurement systems**, v. XVII, n. 2, p. 255–270, 2010.

RAMOS, P. M.; JANEIRO, F. M.; TLEMÇANI, M.; SERRA, A. C.; MEMBER, S. Recent Developments on Impedance Measurements With DSP-Based Ellipse-Fitting Algorithms. **IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT**, v. 58, n. 5, p. 1680–1689, 2009.

RATHORE, T. S. digital measurement techniques. **Digital Measurement Techniques**. Second ed., p.21–30, 2003. Mumbai, India: Alpha Science International Ltda.

RATHORE, T. S. A Simple Circuit for Digital Impedance Measurement. **International Journal of Engineering and Technology**, v. 3, n. November, p. 309–312, 2011.

STAHN, A.; TERBLANCHE, E.; GUNGA, H.-C. **Handbook of Anthropometry: Physical Measures of Human Form in Health and Disease**. 1st ed. New York: Springer Science+Business Media, LLC, 2012.

SUN, R.; SEVILLANO, E.; PERERA, R. Debonding detection of frp strengthened concrete beams by using impedance measurements and an ensemble pso adaptive spectral model. **Composite Structures**, 2015. Elsevier Ltd.

TASSARA, B. P. **Analizador de Amplitude e Fase Cadernos de Saúde Pública**, Oct. 2009. Universidade Federal do Rio de Janeiro - Escola Politécnica.

TI. **SBFA001C - FilterPro™ User ' s Guide**. 2011.

TRIANSTIS, I. F.; DEMOSTHENOUS, A.; RAHAL, M.; HONG, H.; BAYFORD, R. A multi-frequency bioimpedance measurement ASIC for electrical impedance tomography. **2011 Proceedings of the ESSCIRC (ESSCIRC)**, p. 331–334, 2011. IEEE.

WISSENWASSER, J.; VELLEKOOP, M. J.; KAPFERER, W.; LEPPERDINGER, G.; HEER, R. Multifrequency impedance measurement technique for wireless characterization of microbiological cell cultures. **The Review of scientific instruments**, v. 82, n. 11, p. 115110, 2011.

YANG, Y.; WANG, J.; YU, G.; NIU, F.; HE, P. Design and preliminary evaluation of a portable device for the measurement of bioimpedance spectroscopy. **Physiological measurement**, v. 27, n. 12, p. 1293–310, 2006.

ZEEV, E. BEN; ZOLAR, Y.; STEINBERG, E.; ENOSAWA, K.; MARTINEZ, E. Efficient radix-4 FFT on StarCore SC3000 DSPs. **EE Times Automotive DesignLines**, p. 1–8, 2007. Disponível em: <http://www.eetimes.com/document.asp?doc_id=1275484>. Acesso em: 14/08/2014.

ZHANG, J. Q.; XINMIN, Z.; XIAO, H.; JINWEI, S. Sinewave fit algorithm based on total least-squares method with application to ADC effective bits measurement. **IEEE Transactions on Instrumentation and Measurement**, v. 46, n. 4, p. 1026–1030, 1997.

ANEXO A – CÓDIGO MATLAB DO ALGORITMO RADIX-4

```

function S = radix4FFT1_Float(s)
% This is a radix-4 FFT, using decimation in frequency
% The input signal must be floating point
% Works with real or complex input

% Initialize variables and signals
% NOTE: The length of the input signal should be a power of 4: 4, 16, 64, 256, etc.
N = length(s);
M = log2(N)/2;

% Initialize variables for floating point sim
W=exp(-j*2*pi*(0:N-1)/N);
S = complex(zeros(1,N));
sTemp = complex(zeros(1,N));

% FFT algorithm
% Calculate butterflies for first M-1 stages
sTemp = s;
for stage = 0:M-2
    for n=1:N/4
        S((1:4)+(n-1)*4) = radix4bfly(sTemp(n:N/4:end), floor((n-1)/(4^stage)) * (4^stage), 1, W);
    end
    sTemp = S;
end

% Calculate butterflies for last stage
for n=1:N/4
    S((1:4)+(n-1)*4) = radix4bfly(sTemp(n:N/4:end), floor((n-1)/(4^stage)) * (4^stage), 0, W);
end
sTemp = S;

% Rescale the final output
S = S*N;

end

function Z = radix4bfly(x,segment,stageFlag,W)
% For the last stage of a radix-4 FFT all the ABCD multipliers are 1.
% Use the stageFlag variable to indicate the last stage
% stageFlag = 0 indicates last FFT stage, set to 1 otherwise

% Initialize variables and scale to 1/4
a=x(1)*.25;b=x(2)*.25;c=x(3)*.25;d=x(4)*.25;

% Radix-4 Algorithm
A=a+b+c+d;
B=(a-b+c-d)*W(2*segment*stageFlag + 1);
C=(a-b*j-c+d*j)*W(segment*stageFlag + 1);
D=(a+b*j-c-d*j)*W(3*segment*stageFlag + 1);

Z = [A B C D];

end

```

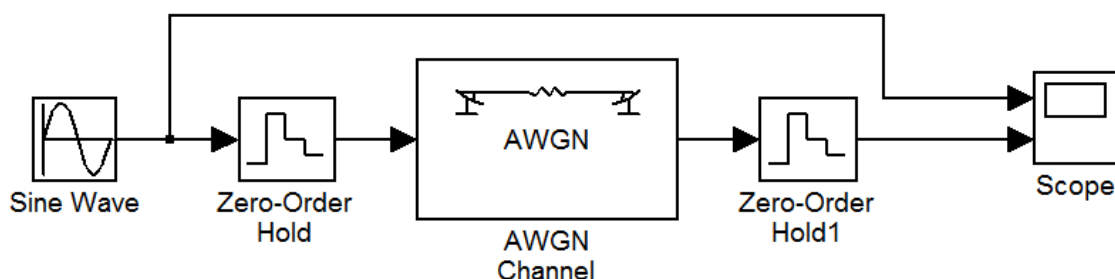

APÊNDICE A – ANÁLISE SNR DOS MÉTODOS TEÓRICOS

Este apêndice tem por objetivo mostrar a sensibilidade de resposta ao ruído dos sete métodos de determinação de fase apresentados nos capítulos 2 e 3. Para a realização desta análise de sensibilidade ao ruído, o ruído foi introduzido apenas no sinal defasado sendo que a função geradora do sinal defasado é descrita pela equação (139).

$$V_{defasada} = 2 \cdot \sin\left(2 \cdot \pi \cdot 50 \times 10^3 \cdot t + \frac{30^\circ}{180^\circ} \cdot \pi\right) \quad (139)$$

A inserção do ruído nos modelos Simulink foi feita substituindo-se a fonte de sinal defasado pelo arranjo experimental mostrado na Figura 96. O bloco “AWGN” é responsável por gerar o ruído branco gaussiano e adicioná-lo ao sinal que entra nesse bloco. O ruído foi ajustado no bloco “AWGN” através do valor da relação sinal ruído (SNR) o qual é introduzido como parâmetro de configuração do bloco. O bloco “AWGN” requer obrigatoriamente o uso de um bloco do tipo “Zero-Order Hold” tanto na sua entrada quanto na sua saída.

Figura 96 - Arranjo para inserção de ruído branco gaussiano.



A resposta de sensibilidade ao ruído foi obtida variando-se o valor de SNR de 0 dB até 100 dB. Os valores utilizados para se plotar as curvas de resposta são os valores obtidos durante os primeiros vinte ciclos em regime permanente para cada modelo. As curvas foram plotadas calculando-se a média desses valores. A Figura 97 mostra a resposta ao ruído obtida pela técnica da figura de Lissajous, a Figura 98 mostra a resposta ao ruído obtida pela técnica do ajuste

de elipse, a Figura 99 mostra a resposta ao ruído obtida pela técnica do demodulador de quadratura, a Figura 100 mostra a resposta ao ruído obtida pela técnica da medição direta digital, a Figura 101 mostra a resposta ao ruído obtida pela técnica PWM, a Figura 102 mostra a resposta ao ruído obtida pela técnica da DFT com algoritmo RADIX-2, a Figura 103 mostra a resposta ao ruído obtida pela técnica da DFT com algoritmo RADIX-4, e a Figura 104 mostra a resposta ao ruído pela técnica do ajuste de senóide.

Figura 97 - Resposta SNR da técnica da figura de Lissajous

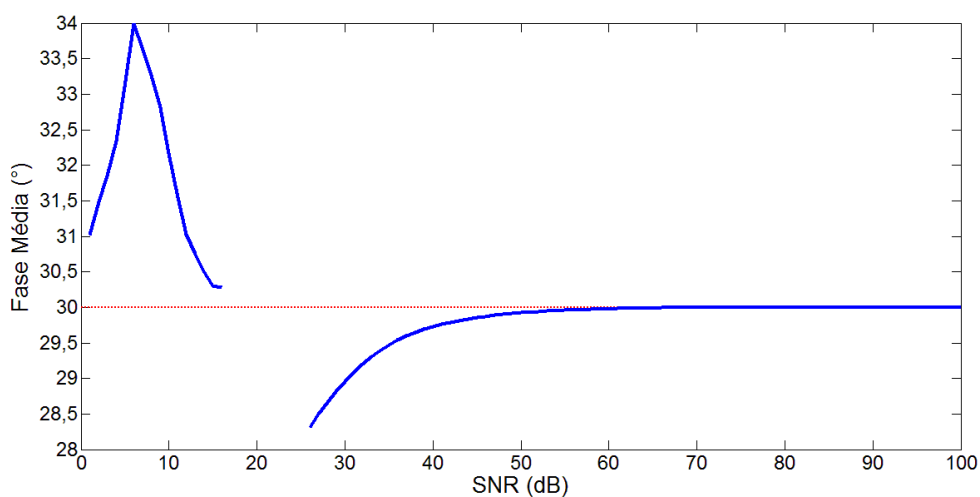


Figura 98 - Resposta SNR da técnica do ajuste de elipse

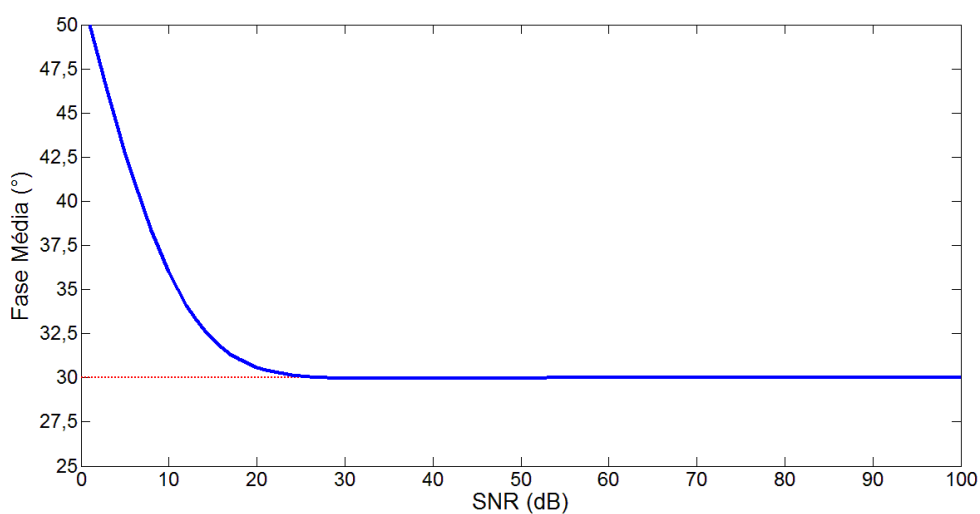


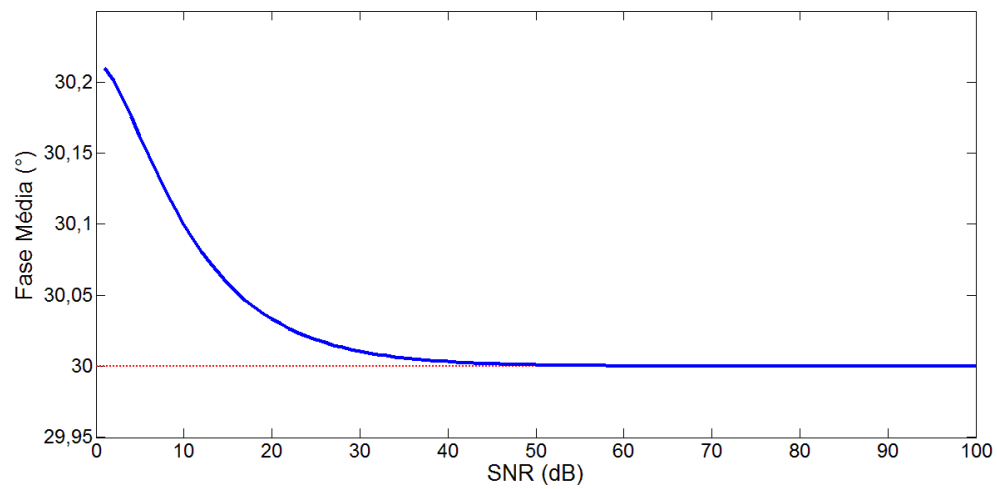
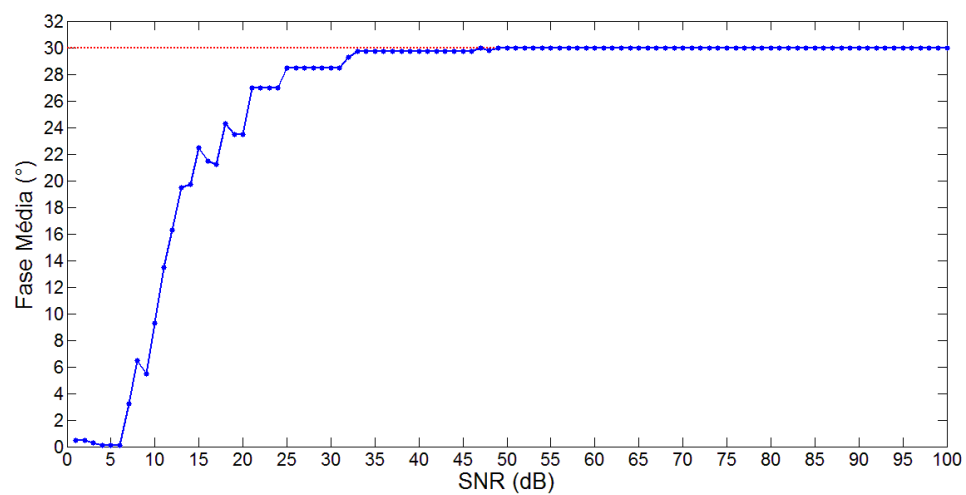
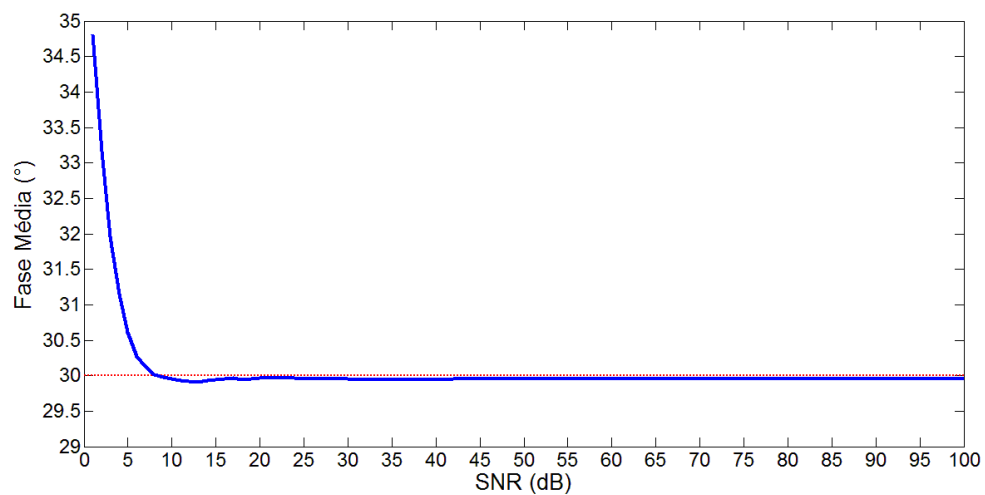
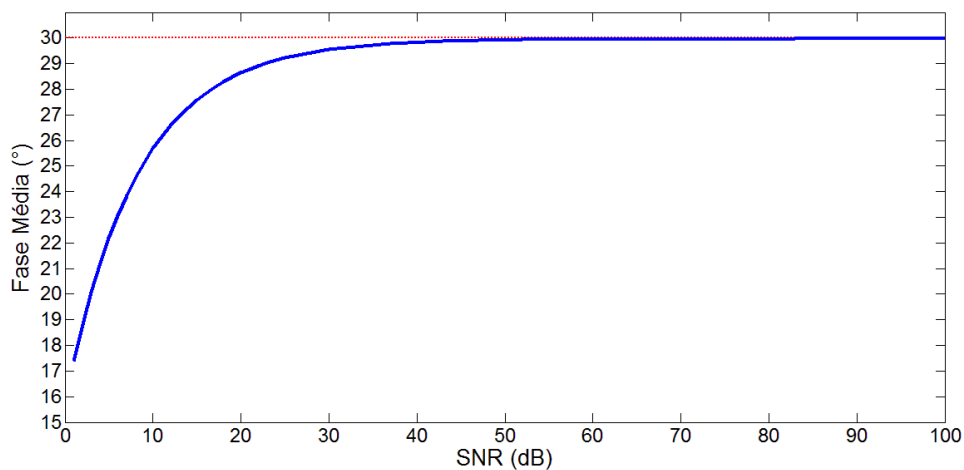
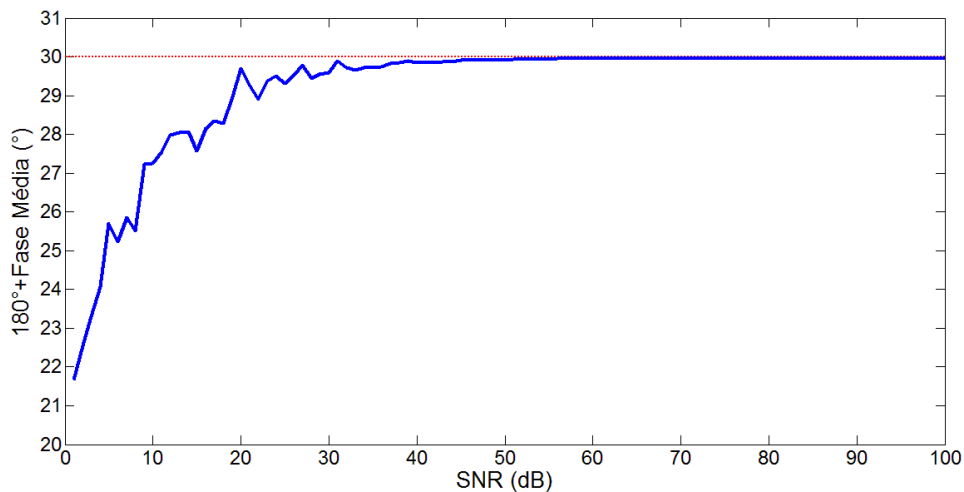
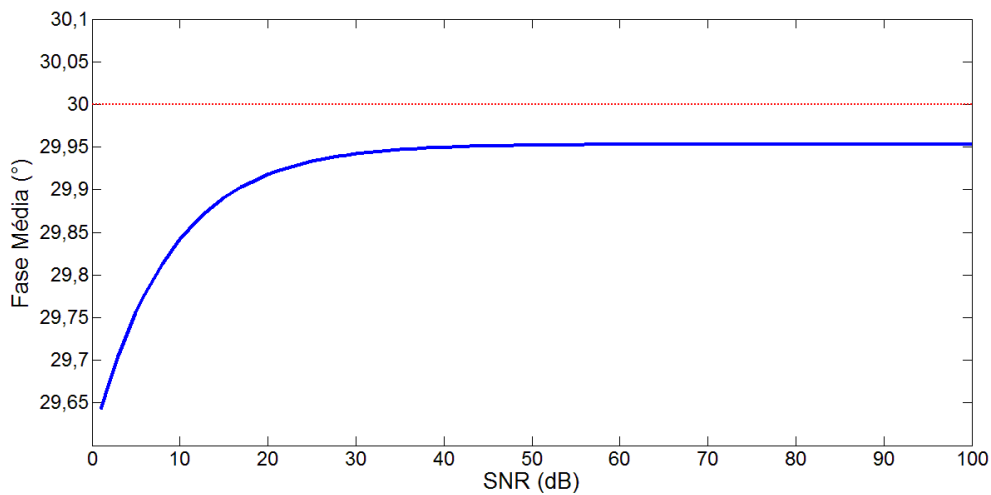
Figura 99 - Resposta SNR da técnica da demodulação de quadratura**Figura 100** - Resposta SNR da técnica de medição direta digital**Figura 101** – Resposta SNR para a técnica PWM

Figura 102 - Resposta SNR para a técnica DFT - Radix-2**Figura 103 - Resposta SNR para a técnica da DFT - Radix-4****Figura 104 - Resposta SNR para a técnica do ajuste de senóide**

A técnica da figura de Lissajous tem uma resposta próxima ao valor teórico a partir de 55 dB enquanto que a técnica do ajuste de elipse se aproxima do valor teórico a partir de 30 dB. A demodulação de quadratura se aproxima do valor teórico a partir de 42 dB, a técnica da medição direta digital se aproxima do valor teórica a partir de 49 dB, e a técnica da modulação de largura de pulso (PWM) se aproxima do valor teórica a partir de 43 dB. A técnica da transformada discreta de Fourier com algoritmo Radix-2 se aproxima do valor teórico a partir de 52 dB enquanto que a mesma técnica com o algoritmo Radix-4 se aproxima do valor teórico a partir do valor de 56 dB. A técnica do ajuste de senóide se aproxima do valor teórico a partir de 56 dB.

As técnicas com a menor variação de resposta foram a técnica da demodulação de quadratura ($0,2^\circ$) e a técnica do ajuste de senóide ($0,35^\circ$). A técnica da figura de Lissajous teve variação de resposta de 4° enquanto que a técnica do ajuste de elipse teve variação de resposta de 20° . A técnica da medição direta digital teve variação de 30° . A técnica de PWM teve variação de resposta de 5° . A técnica da DFT com algoritmo Radix-2 teve variação de resposta de 13° enquanto a mesma técnica com algoritmo Radix-4 teve variação de resposta de 9° .

A tabela 13 compara os valores de variação de resposta e o valor SNR a partir do qual a resposta se aproxima do valor teórico de fase para as sete técnicas testadas.

Tabela 13 - Comparação de resposta SNR

Técnica	SNR para valor teórico (dB)	Variação de resposta ($^\circ$)
Figura de Lissajous	55	4
Ajuste de Elipse	30	20
Demodulação de Quadratura	42	0,2
Medição Direta Digital	49	30
PWM	43	5
DFT com Radix – 2	52	13
DFT com Radix – 4	56	9
Ajuste de Senóide	56	0,35